ANL/NSE-21/79

Nek5000: improvements in the available RANS models, meshing, tutorials, and training

Nuclear Science and Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

ANL/NSE-21/79

Nek5000: improvements in the available RANS models, meshing, tutorials, and training

prepared by

Dillon Shaver[1], Aleks Obabko[2], Ananias Tomboulides[3,4], Jun Fang[1], Haomin Yuan[1], Yiqi Yu[1], Sierra Tutwiler[1,5], Dezhi Dai[1], Nadish Saini[1], and Christopher Boyd[6]

[1]Nuclear Science and Engineering Division, Argonne National Laboratory
[2]Computational Science Division, Argonne National Laboratory
[3]Mathematics and Computer Science Division, Argonne National Laboratory
[4]Aristotle University of Thessaloniki
[5]Virginia Commonwealth University
[6]U.S. Nuclear Regulatory Commission

September 30, 2021

Abstract

This year, the Nuclear Energy Advanced Modeling Simulation program (NEAMS) thermal-hydraulics report for Nek5000 NRC- and verification and validation (V&V)-driven development focuses on following areas of code application and improvement. First we have continued improvements of RANS modeling capabilities in Nek5000 including improved k-tau model focusing mostly on wall-function initial implementation with spectral element method (SEM) and initiating investigation of an alternative approach XSEM that greatly reduces discretization errors.

Second, in a close collaborative effort with the U. S. Nuclear Regulatory Commission (NRC) staff, we have continued V&V efforts for the HYMERES-2 project using an OECD/NEA sponsored test series in the PSI PANDA facility. This year's focus of ANL-NRC collaboration involves Nek5000 setups and validation for a range of problems relevant to and including the HYMERES-2 benchmark from PSI. The primary outcome of this year's effort is a more efficient geometry and inlet modeling simplification after a careful sensitivity study of the inlet profiles and pipe geometries. The resulting modeling choice of a short recycling/fully-developed turbulent inlet exceeding some of the experimental measurement uncertainty estimates. This finding simplifies the next step of the cross-V&V HYMERES-2 project of full vessel geometry LES whose higher resolution cases are underway together with setups of the transient heat and mass transfer cases including buoyancy effects. In addition, the ANL team continue to provide assistance to the NRC staff in the form of Nek5000 application support in general and on the use of the HPC platforms of ALCF and INL in particular. This supports the NRC's assessment of Nek5000 for use with the NRC Blue CRAB code suite.

Third, we report on the initial implementation of a quadratic tet-to-hex meshing capability. This capability allows for a robust meshing capability that is conformal to the problem geometry to 2nd order. It is tested for tet-to-hex and wedge-to-hex and applied to the reactor pressure vessel downcomer for the ROCOM facility.

Finally, we report on enhancements to the documented tutorials and describe training activities conducted this year. The conjugate heat transfer tutorial was significantly edited for better clarity in response to user input and feedback. Additionally, a new tutorial for laminar flow in a channel was added. This is intended as a first case for beginning users and covers basic problem setup with prescribed boundary conditions. A training session was hosted virtually in response to a request from the Microreactors program at INL. Attendees were guided through two example cases with Nek5000 and provided with a primer on the use of Gmsh.

Contents

List of Figures

List of Tables

1 Introduction

This year, the Nuclear Energy Advanced Modeling Simulation program (NEAMS) thermal-hydraulics report for Nek5000 [1] verification and validation (V&V) -driven development focuses on three areas of code application and improvement. First, following industry preferences, we have continued improvements of RANS modeling capabilities in Nek5000 (and its GPU variant NekRS) including improved k-tau model with and without wall-functions and initial investigation of an alternative approach of eXtended/enriched spectral element method (XSEM). Second, we have continue assisting and collaborating with the U. S. Nuclear Regulatory Commission (NRC) staff with Nek5000 setups and validation for the Hydrogen Mitigation Experiments for REactor Safety (HYMERES-2) benchmark. Lastly, we report the initial improvements of quadratic tet-to-hex meshing implementation and further training activities and tutorials.

With the U. S. nuclear industry on the cusp of deploying the next-generation of power reactors, the NEAMS program is charged with providing the next-generation of modeling and simulation tools. The objective of this work is to assess capabilities in addressing the needs that have been identified as important to both the DOE-NE Advanced Reactor Technologies program (ART) and the nuclear industry. The focus here is on Nek5000, an open-source, highly scalable computational fluid dynamics (CFD) code based on the spectral element method. Nek5000 has traditionally been used to provide accurate reference solutions produced with its high-fidelity capability, typically LES, that could be further used for benchmarking and improving uncertainty estimation for lower-fidelity, faster-turn-around approaches. By building on that pedigree, this work aims to extend the capabilities of Nek5000 to make it more practical for use on problems of relevance to the industry and the NRC.

In a close collaborative effort with the NRC staff, we have continued V&V activity using the Nuclear Energy Agency of the Organization for Economic Co-operation and Development (OECD/NEA) sponsored testing in the PANDA facility. Located at the Paul Scherrer Institute (PSI) in Switzerland, the PANDA facility is a multi-compartment, large-scale thermal-hydraulics test rig that has been used in numerous tests and benchmarks. Recent tests have been focused on providing data for validation of codes for prediction of distribution of buoyant gases including hydrogen during Fukushima-related accident events. Data from these tests has been used as the basis for comparison with CFD results using URANS and LES models in Nek5000.

Prevously, the initial meshing and preliminary LES tests kicked off the ANL-NRC collaboration. This collaboration is directly supporting the longer term goals of the NRC related to the improvement of lower-fidelity fast-turn-around URANS based turbulence modeling capabilities. In particular, the current model improvement effort is focused on validating against erosion of an air-helium stratified layer as investigated using the previous OECD/NEA PANDA benchmark [2, 3] and current HYMERES-2 project [4].

The primary focus of this year's efforts of the collaboration was an improvement in modeling the benchmark inlet conditions and longer term flow evolution simulation. It was confirmed that the conditions are sensitive to simplifications in the meshing, geometry, and the upstream level of turbulence. Taking into the account the computational efficiency, and thus time-to-solution of the full benchmark geometry problem, the improvement of meshing has been a target. In addition, the

ANL team has continued to provide assistance to the NRC staff in the form of Nek5000 application support during the NRC's assessment of the solver usage to support the NRC's Comprehensive Reactor Analysis Bundle (CRAB) code suite.

Another area of the V&V-driven development of Nek5000 that is/will be important to nuclear industry and NRC is improvement of URANS implementation in the code. Several of the applications of interest to NEAMS can be addressed through RANS modeling including. e.g. liquid fuel molten salt fast reactors (MSFRs). The RANS models recently implemented in Nek5000 were based on the $k-\omega$ model[5]. A significant development during the past year was the improvements and further tests of newly implemented $k-\tau$ model, which was originally developed by Kalitzin et al.[6, 7] as an alternative implementation of the standard $k-\omega$ model. In contrast to the original model, in which the ω equation contains terms that become singular close to walls, all terms in the k and τ equations reach a finite limit which facilitates their numerical implementation. Moreover, this model does not rely on the wall-distance function or its derivatives and is better suited for wall-function implementation that was also the focus of this year's development.

We investigated and tested various ways to increase the stability, accuracy and robustness of our RANS approaches. This includes the wall function formulation in SEM that presents peculiar challenges due to non-local structure of discretization within an element. Note that leveraging the support of the Exascale Computing Project (ECP) allowed us to implement some of the improvements in the GPU version of Nek5000, NekRS, that we have had an initial discussion with the NRC on.

To facilitate meshing of advanced reactor components, a significant upgrade to our SEM meshing capability was made via an initial implementation of a quadratic tet-to-hex method. This improves the ability to create accurate meshes of complex geometries, as previously this capability was limited to producing only first-order accurate meshes.

In an effort to make Nek5000 more accessible to users, we also made various improvement to the documented tutorials and hosted a virtual training session. A summary of the contents of the tutorials and the topics covered by the training are reported here along with feedback provided by the training attendees.

The report is organized as follows. The URANS implementation improvements in Nek5000 are described in Section 2. Section 3 describes further NRC-ANL collaborative work on Nek5000 application to the OECD/NEA HYMERES-2 relevant geometries. The meshing improvements and tutorials with training activities are reported in Section 4 and 5, respectively. We conclude in Section 6 with a brief summary and outline of the future work.

2 Improvements to RANS Modeling

2.1 The $k - \tau$ model

RANS models describe the turbulent properties of incompressible flows with

$$k = \frac{\langle u'^2 \rangle + \langle v'^2 \rangle + \langle w'^2 \rangle}{2}, \tag{1}$$

where u', v', and w' are the fluctuation components of the velocity vector around the ensemble-averaged mean velocity vector $\mathbf{v} = (u, v, w)$, governed by

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla p + \nabla \cdot \left[(\mu + \mu_t) \left(2\mathbf{S} - \frac{2}{3} Q \mathbf{I} \right) \right], \tag{2}$$

where

$$\mathbf{S} = \mathbf{\frac{1}{2}} \left(\nabla \mathbf{v} + \nabla \mathbf{v}^{\mathbf{T}} \right), \tag{3}$$

μ is the molecular viscosity and μ_t is the turbulent viscosity, with the continuity equation for incompressible flow being

$$Q = \nabla \cdot \mathbf{v} = 0. \tag{4}$$

The divergence of velocity Q can be nonzero in the case of reactive or multiphase flows.

We have implemented and tested several RANS approaches in Nek5000, in the frame of the spectral element method (SEM), including a regularized version of the $k - \omega$ model [8, 9, 5]. A significant recent development was the implementation and testing of the $k - \tau$ model, which was originally developed by Kalitzin et al. [6, 7] and by Speziale et al. [10] as an alternative implementation of the standard $k - \omega$ model. Details of this implementation and its verification in Nek5000 are available in [11]. In contrast to the original form of the $k - \omega$ model, in which the ω equation contains terms that become singular close to wall boundaries, all terms in the right-hand side of the k and τ equations reach a finite limit at walls and do not need to be treated asymptotically; that is, they do not require regularization for numerical implementation. In this work the $k - \tau$ model is used. The equations for k and τ are derived from the $k - \omega$ equations by using the definition $\tau = 1/\omega$:

$$\begin{aligned}
\frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho k \mathbf{v}) &= \nabla \cdot \left[(\mu + \frac{\mu_t}{\sigma_k}) \nabla k \right] + P - \rho \beta^* \frac{k}{\tau}, &(5)\\
\frac{\partial(\rho \tau)}{\partial t} + \nabla \cdot (\rho \tau \mathbf{v}) &= \nabla \cdot \left[(\mu + \frac{\mu_t}{\sigma_\omega}) \nabla \tau \right] - \gamma \frac{\tau}{k} P + \rho \beta - 2 \frac{\mu}{\tau} \left(\nabla \tau \cdot \nabla \tau \right), &(6)
\end{aligned}$$

where P is the rate of production of TKE. The last term in the τ equation was implemented in the form proposed by [12], as

$$S_\tau = 2\nu \left(\nabla \tau \cdot \nabla \tau \right) / \tau = 8\nu \left(\nabla \tau^{1/2} \cdot \nabla \tau^{1/2} \right). \tag{7}$$

Looking closer into the scaling of all the terms appearing in the right-hand side of the k and τ equations, one can observe that near walls, the two main terms of the k equation balance each other:

$$Y_k = \rho\beta^* \frac{k}{\tau} \approx \mu\nabla^2 k, \tag{8}$$

whereas the dissipation and diffusion terms in the τ equation behave as

$$Y_\tau = \rho\beta \to \rho\beta \tag{9}$$

$$\nabla \cdot (\mu\nabla\tau) \to \frac{1}{3}\rho\beta \tag{10}$$

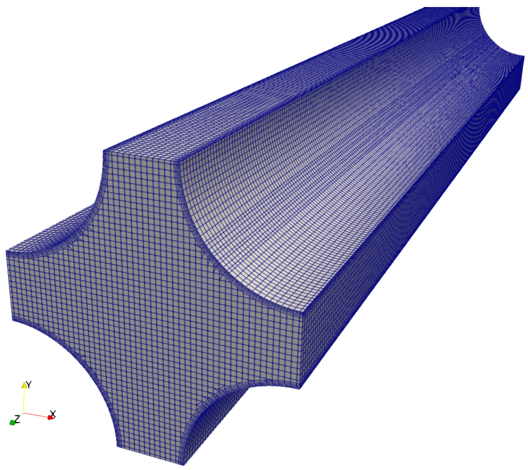
$$S_\tau = 2\frac{\mu}{\tau}(\nabla\tau \cdot \nabla\tau) \to \frac{4}{3}\rho\beta. \tag{11}$$

2.2 Comparison with OpenFOAM

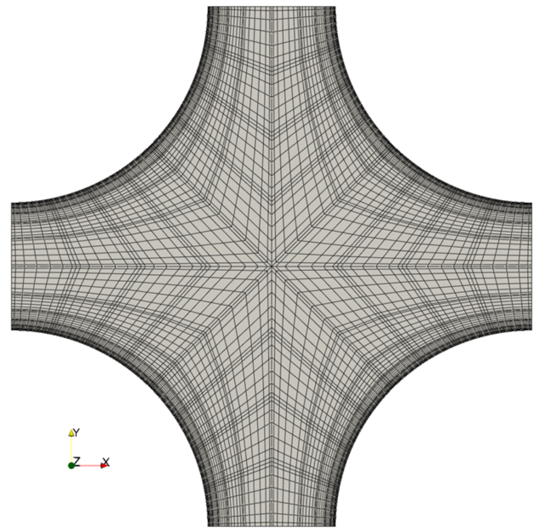
The original implementation of the $k-\tau$ model with implicitized source terms was verified with comparisons to flows in a backward facing step and over an airfoil [11]. Additionally, the $k-\tau$ model was compared against results with the regularized $k-\omega$ model. However, unlike the original implementation of the $k-\omega$ model [5], it has not previously been compared to model implementations in other codes. To provide an additional point of comparison for the $k-\tau$ model implementation in Nek5000, A simulation was performed for a single subchannel between 4 fuel pins at a Reynolds number of 50,000. Results from the wall-resolved $k-\tau$ model were compared to a similar $k-\omega$ based model available in OpenFOAM version 2012 [13]. By comparing the Nek5000 results to those obtained with a different code provides evidence that the model has been implemented successfully, sometimes referred to as cross-code verification.

A mesh independence study was performed for the case in OpenFOAM. This was done through progressive refinements to the mesh until the solution was determined to no longer change with further refinement and through monitoring of the near-wall y^+ values. The final mesh used consisted of 550,560 elements with average and maximum y^+ values of 0.56 and 0.89 respectively. Similarly for the Nek5000 result, p-type refinement was performed using 5th and 7th order polynomials and the near-wall y^+ values were monitored with average and maximum values of 0.846 and 0.883 respectively for the 7th order mesh. Both meshes are shown in Figure 2.1.

Comparison of the results for the velocity profile are presented as a 2D color map in Figure 2.2 and as a line plot across the channel diagonal in Figure 2.3. The two profiles shown in Figure 2.2 are qualitatively similar, both showing typical RANS-type velocity distributions. For a more quantitative comparison, the line plots in Figure 2.3 can be seen to have very similar profiles. There are some minor differences, with the Nek5000 result peaking at slightly higher velocity at the channel center compared to the OpenFOAM result. While it is difficult to determine if these differences are attributable to the differences in the underlying models, i.e. $k-\tau$ vs. $k-\omega$, or the numerical method, they are small enough to conclude that the wall resolved $k-\tau$ model has been implemented consistently.

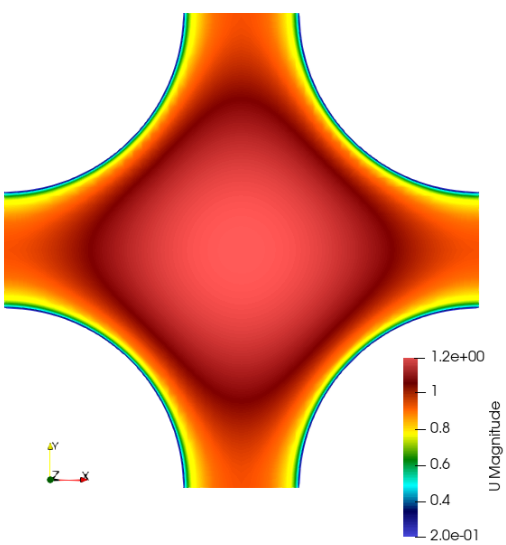


(a) The mesh used for the OpenFOAM simulation

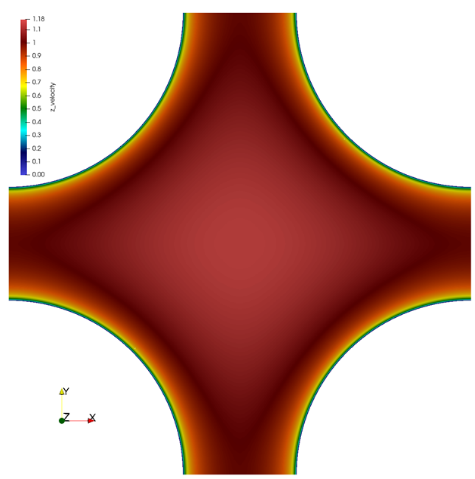


(b) Cross section of the mesh used for Nek5000 simulation

Figure 2.1: The meshes used for the comparison case between Nek5000 and OpenFOAM



(a) OpenFOAM



(b) Nek5000

Figure 2.2: Comparison of the velocity distribution obtained using OpenFOAM and Nek5000

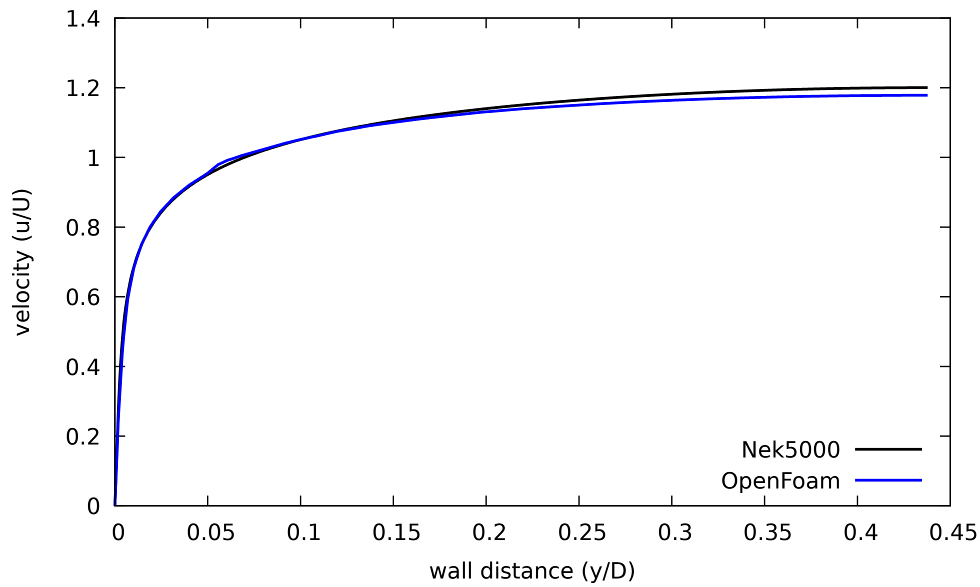


Figure 2.3: Comparison of velocity profiles across the subchannel diagonal

2.3 Wall function implementation

Here we focus on the implementation of the wall modeled version in the context of wall functions. To avoid the need for resolution of strong velocity gradients, wall functions are typically applied. For the wall modeled version we follow the approach of Grotjans and Menter [14] and Kuzmin et al. [15] which is appropriate for finite element methods based on the weighted residual approach and the variational formulation.

At solid boundaries the normal component of the velocity is set equal to zero whereas the tangential component is permitted to have a slip in turbulent flow simulations. The traction boundary conditions imposed on the tangential velocity are based on the boundary conditions for k and τ at the boundary and the law of the wall. The implementation of this boundary condition for the velocity in Nek5000 is performed using the full stress formulation [16] and allows for boundaries that are curved and not aligned with any of the axes.

The exact form of the traction boundary condition for the tangential velocity $\mathbf{u}_t = \mathbf{u} - \mathbf{n}(\mathbf{n} \cdot \mathbf{u})$ for the case that the normal to the boundary direction is aligned with the y-direction is obtained as follows:

$$\tau_w = \nu_t \frac{\partial |\mathbf{u}_t|}{\partial y} \tag{12}$$

for the log-law

$$u^+ = \frac{|\mathbf{u}_t|}{u_\tau} = \frac{1}{\kappa} ln \left(E y^+ \right) \tag{13}$$

where $\kappa = 0.41$ is the von Karman constant and $E = 9$. The eddy viscosity at the boundary is

$\nu_t = \kappa \nu y^+$ and τ_w is given by:

$$\tau_w = (\kappa \nu y^+) \frac{\partial |\mathbf{u}_t|}{\partial y} = u_\tau^2 \tag{14}$$

Thus the tangential velocity gradient in the normal to the wall direction is given by

$$\frac{\partial |\mathbf{u}_t|}{\partial y} = \frac{\tau_w}{\nu_t} = \frac{u_\tau^2}{(\kappa \nu y^+)} \tag{15}$$

According to Grotjans and Menter [14] an explicit relation for the friction velocity u_τ which is required to evaluate the tangential stress for the momentum equations, is:

$$u_\tau = \max \left(u^*, \frac{|\mathbf{u}_t|}{u^+} \right) \tag{16}$$

where $u^* = C_\mu^{1/4} k^{1/2}$ and the value of k at a location inside the log layer is given by:

$$k = \frac{u_\tau^2}{C_\mu^{1/2}} \tag{17}$$

The momentum flux at the boundary for the tangential velocity component, which appears in the boundary integral term after applying the variational formulation and for the general case is given by

$$2(\nu + \nu_t)(\mathbf{n} \cdot \mathbf{S}) = \tau_w \frac{(\nu + \nu_t)}{\nu_t} \frac{\mathbf{u}_t}{|\mathbf{u}_t|} = u_\tau u^* \left(1 + \frac{1}{\kappa y^+} \right) \frac{\mathbf{u}_t}{|\mathbf{u}_t|} = u^* \frac{\mathbf{u}_t}{u^+} \tag{18}$$

In this approach, the boundary of the computational domain is not located exactly at the wall but at a finite, distance from the wall corresponding to a fixed value of y^+. Strictly speaking, this implies that a boundary layer of width y (corresponding to the specified value of y^+) should be removed from the computational domain; however, it is assumed that this width is very small at high Reynolds numbers and can be considered negligible, so that the equations can be solved in the whole domain with wall functions prescribed on the boundary.

Since the choice of y^+ is rather arbitrary, it is possible to define its value for example as the point where the logarithmic layer meets the viscous sublayer; it can also be defined as a specific location inside the logarithmic layer. In any case the momentum flux at the boundary is based on the value of u_τ obtained from the law of the wall. In this work we specified a value of y^+ which is well inside the log layer and it ranged between $30 < y^+ < 200$.

Following [14] we impose a zero Neumann boundary conditions for k, which can be derived from Eq. (17), i.e. $\partial k / \partial y = 0$. For τ we impose a Dirichlet boundary condition, using Eq. (17) for k:

$$\tau = \frac{\nu_t}{k} = \frac{\kappa \nu y^+}{k} \tag{19}$$

The type of wall functions used here which forces the normal velocity at a wall boundary to be zero but allows a slip for the tangential velocity component is not well posed at sharp corners of any angle [17]. This can create problems with mass conservation as well as numerical instability due to noise. This problem limits the applicability of this wall function approach to simple geometries without sharp corners.

To resolve this issue we chose an approach in which we do not use wall function boundary conditions at faces of spectral elements which are immediately adjacent to corners in 2D or corners and edges in 3D and instead we use wall boundary conditions at those points. This means that all velocity components are zero at those corner faces and k and τ are both equal to zero.

The approach described above is possible for the $k - \tau$ model where both k and τ approach zero at walls. In contrast, it would not be possible for the $k - \omega$ model because ω becomes infinite at wall boundaries. The boundary conditions at the corner faces are converted to wall-type at a pre-processing step in the beginning of the simulation. We found this approach to be robust and to allow the wall modeled RANS simulations of complex flows at high Reynolds numbers without the need for additional resolution close to walls.

Enriched XSEM

We are currently investigating an alternative approach to compute the flow in the near wall region using RANS or LES in a cost-effective and accurate way. This approach is based on the concept of function enrichment [18], [19] and the idea is to enrich the polynomial spectral element (SEM) approximation space with additional shape functions that include log-law like profiles to reduce resolution requirements [20], [21]. For the sake of brevity the term XSEM will be used to describe this approach (for enriched or eXtended SEM).

The velocity profile is modeled using these additional "wall functions" inside the elements adjacent to walls and the no-slip boundary condition is satisfied for all velocity components. This approach enables the use of coarse meshes in the vicinity of walls while the method can still accurately account for pressure gradients and non-equilibrium effects. The numerical method will automatically find the optimal solution as a linear combination of the "wall function" , which enables the accurate representation of the high gradient at the wall, and the Legendre Lagrangian interpolants. As of now we have investigated the implementation of the XSEM approach to solve the convection-diffusion equation for a model problem desribed below, which has an analytical solution:

$$-\nu\frac{d^2u}{dx^2} + c\frac{du}{dx} = 1, \quad u(0) = u(1) = 0 \tag{20}$$

The analytical solution of (20) is:

$$u(x) = \frac{1}{c}\left(x - L\frac{e^{c(x-L)/\nu} - e^{-cL/\nu}}{1 - e^{-cL/\nu}}\right) = \frac{L}{c}\left(\tilde{x} - \frac{e^{Pe(\tilde{x}-1)} - e^{-Pe}}{1 - e^{-Pe}}\right) \tag{21}$$

where $\tilde{x} = x/L$, the convection velocity $c = 1$, the domain length $L = 1$, the diffusion coefficient $\nu = 0.01$ and the Peclet number $Pe = cL/\nu = 100$. The solution has a boundary layer at $x = 1$ with a thickness that depends on the value of Pe. Equation (20) is discretized using the SEM basis, enriched inside the last spectral element (which includes $x = 1$) with a non-polynomial shape function given by:

$$h(x) = exp(\frac{c(\tilde{x}-1)}{\nu}) \tag{22}$$

The overall numerical approximation of the solution including the enrichment is given by:

$$u_h = \sum_{j=0}^{N} u_j h_j - u_N h_N h \tag{23}$$

where h_j is the Legendre Lagrangian inteprolant of collocation point j, u_N is the Galerkin coefficient corresponding to point $x = 1$, and h_N is the Legendre Lagrangian inteprolant of collocation point $j = N$. It is important to note that in the enriched space, u_N is not the actual value of the solution u at the last node $j = N$, i.e. at $x = 1$, so it does not satisfy the homogeneous boundary condition $u(1) = 0$. Instead, it is a Galerkin coefficient which multiplies the enrichment term and which when added to the standard SEM expression, it forces the solution to satisfy the homogenous boundary condition. The discretization above results in a system of the following form:

$$(A_h - A_x)\,\mathbf{u} + (C_h - C_x)\,\mathbf{u} = (B_h - Bx)\,\mathbf{I} \tag{24}$$

where A_h, C_h and B_h are the standard SEM forms of the stiffness matrix, convection operator and mass matrix, respectively and A_x, C_x and B_x are the corresponding matrices for the enriched parts. System (24) can be inverted directly to obtain the numerical solution $\mathbf{u}$ but it can also be time-marched to steady-state starting from an arbitrary initial condition. We have verified that it is possible to perform the latter by explicitly extrapolating the enriched terms. This approach allows the main structure of the Nek5000 operators and routines to remain the same while the enriched terms can be added to the explicitly treated right-hand side. The solution of the above problem is shown in figure (2.4) using only 3 spectral elements in x. In general by using enrichment, the overall error is significantly lower than the non-enriched spectral element solution on the same mesh. It should be noted that due to the non-polynomial nature of the enrichment shape function (22), high-order quadrature has to be used for the accurate evaluation of all integrals appearing in the enriched terms.

In work underway, we are implementating the enriched XSEM method for the RANS equations, using shape functions that include log-law like profiles to enrich the approximation space in order to reduce resolution requirements We are currently testing this implementation for RANS in parallel channel flow. In future work we plan to investigate the use of enriched wall models also for hybrid RANS-LES approaches.

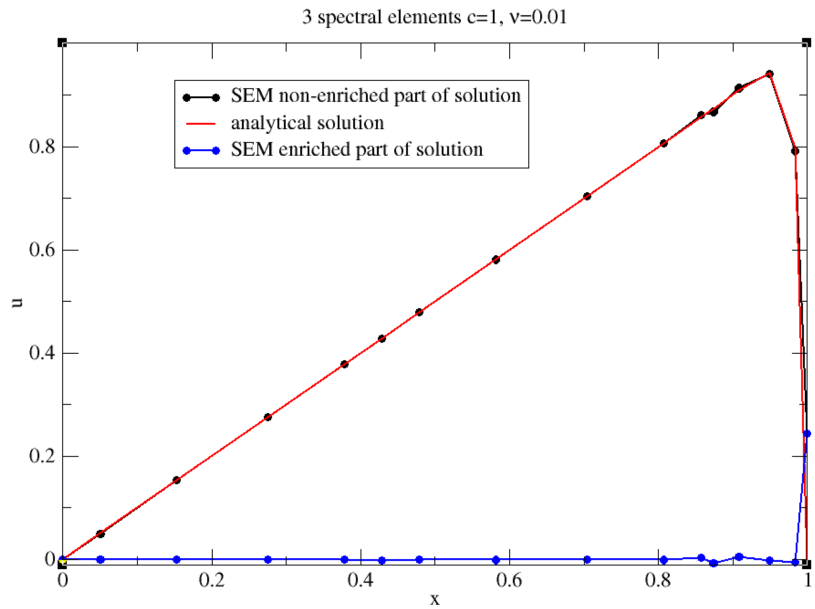


Figure 2.4: XSEM solution for the convection-diffusion model problem

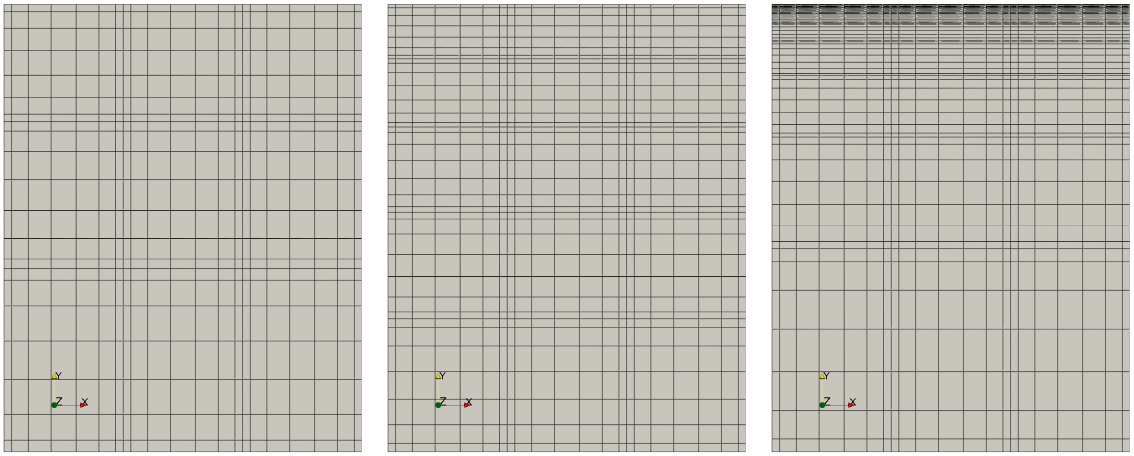
2.4 Results with wall functions

Channel Flow

As a starting point, the $k - \tau$ RANS model with wall functions has been used to simulate flow in a periodic channel. This represents the simplest possible case with well known solutions for comparison. A heated case with $Re = 50,000$ based on the channel half-width and $Pr = 1$ has been simulated using both the wall modeled and wall resolved approaches. Additionally, the wall modeled approach has been simulated on multiple computational meshes.

An interesting feature of the wall function implementation as described above is the choice of a specified value of y^+. For all of the described cases, a value of $y^+ = 100$ was chosen as this represents a value well within the logarithmic layer. Because this value is assigned as part of the boundary condition, the predictions of the wall model are independent of the next-to-wall node spacing. This is in contrast to typical implementations in the finite volume method where the next-to-wall y^+ is part of the solution and must be monitored carefully. For a high-y^+ (high-Re) implementation, this value must remain within the logarithmic layer. This can lead to "over-refinement" of the computational mesh in areas where the next-to-wall y^+ falls into the transition layer or even the laminar sublayer.

Three computational meshes for the channel flow case are shown in Figure 2.5, coarse, medium, and fine. Note that the fine mesh was designed for use with a wall resolved model and has a much more aggressive geometric growth factor. The fine mesh represents the minimum required resolution for the wall resolved model.



(a) coarse mesh (3×3) (b) medium mesh (3×5) (c) fine mesh (3×8)

Figure 2.5: comparison of the meshes used for the channel flow simulation with the wall modeled approach, the wall boundary is at the top of the domain

Comparisons of the predicted velocity, temperature, and turbulent kinetic energy profiles are shown in Figure 2.6 in both standard and wall coordinates. For all three mesh refinement levels, the results for the wall modeled approach are practically identical. As the finest mesh is refined to an equivalent y^+ value of < 1, this shows that the implemented model demonstrates true mesh convergence and is free of any possible “over-refinement” constraints. To quantify this, values for the Darcy friction factors and Nusselt numbers are computed and presented in Table 2.1. The values for the wall modeled approach agree quite well with the wall resolved approach, and practically no difference is observed between meshes.

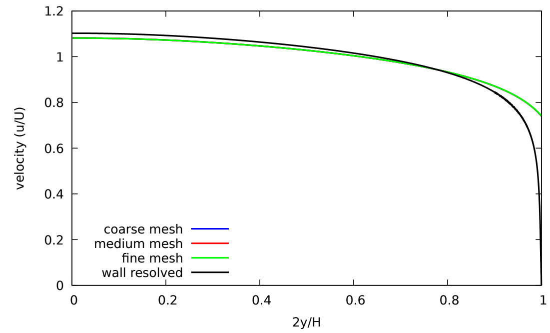
Table 2.1: Comparison of friction factors and Nusselt numbers to the wall resolved approach

case	friction factor	Nusselt number
wall resolved	0.00409	108.20
coarse	0.00393	101.51
medium	0.00392	101.53
fine	0.00392	101.53

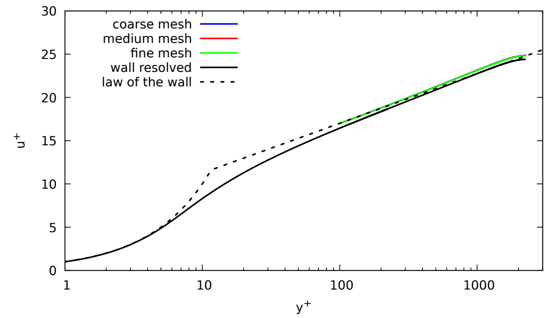
While it can be seen that the velocity and temperature profiles match well for the bulk of the flow, there are some differences observed near the wall, however. This is expected as the wall function approach does not resolve the details near the wall. From the wall coordinate plots, it can be seen that the two models agree quite well from the logarithmic layer into the bulk of the flow. In particularly, it is observed that both modeling approach agree well with the law of the wall, given by Eq. (13) and

$$T^+ = y^+_{CSL} + \frac{Pr_t}{\kappa} \ln(\frac{y^+}{y^+_{CSL}}), \tag{25}$$

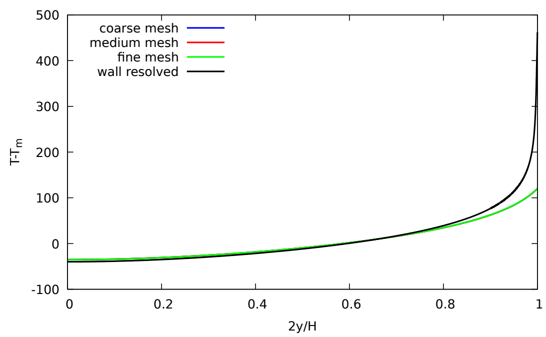
where the wall distance of the conduction sublayer is assumed $y^+_{CSL} = 11.6$. For the turbulent kinetic energy, the profile is matched across nearly all of the channel, except for the steep gradient



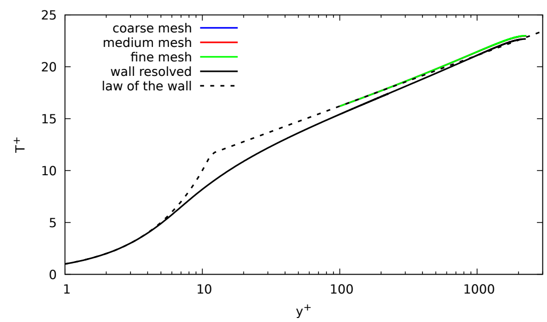
(a) Velocity (u/U)



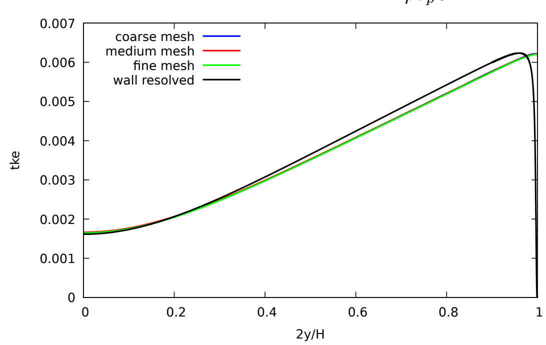
(b) u^+ (u/u_τ)



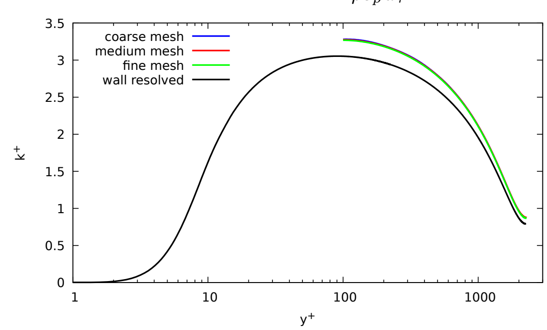
(c) Temperature $(T - T_b)\frac{q''}{\rho c_p U}$



(d) T^+ $(T - T_b)\frac{q''}{\rho c_p u_\tau}$



(e) Turbulent kinetic energy (k/U^2)



(f) k^+ (k/u_τ^2)

Figure 2.6: Velocity, temperature, and TKE profiles for the wall modeled approach on various meshes compared to the wall resolved approach for channel flow

near the wall. This difference is clearly seen in the plot of k^+.

With the coarse mesh, the wall modeled approach is able to predict friction factor and Nusselt number to within a few percent difference of the wall resolved approach using less than half the required number of elements. While channel flow is an ideal use case for wall functions, the potential gains in required number of elements can be significantly compounded for complex geometries.

Pipe Flow

A series of cases of pipe flow have been simulated with the wall modeled $k-\tau$ model at a range of Reynolds numbers. The use of pipe flow as a benchmark has been chosen due as it is again, a simple and well studied problem. Additionally, pipe flows are ubiquitous in engineering applications. This demonstrates the behavior of the model in predicting heat transfer and pressure drop over a range of conditions that would be computationally prohibitive for either a wall resolved model or a full LES model. Results for friction factor and Nusselt number for these cases are compared to the Prandtl and Dittus-Boelter correlations respectively.

$$\frac{1}{\sqrt{f}} = 2\log_{10}\left(Re\sqrt{f}\right) - 0.8 \tag{26}$$

$$Nu = 0.023Re^{0.8}Pr^{0.4} \tag{27}$$

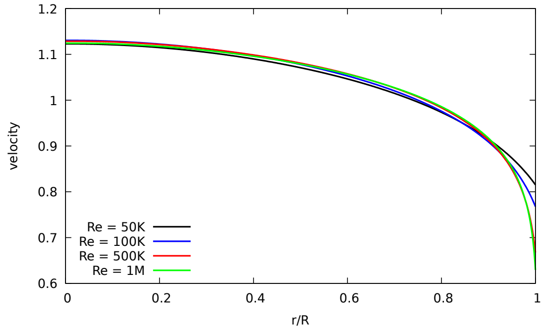
For the sake of simplicity, $Pr = 1$ was chosen.

Profiles of the velocity, temperature, and turbulent kinetic energy are presented in Figure 2.7. As was previously described in section 2.3, with increasing Reynolds number, the domain error associated with prescribing a y^+ value becomes smaller. This is observed in the figure as a sharpening of the velocity and temperature profiles. The value of velocity on the boundary decreases, while the value of the temperature increases. Dimensionless profiles of turbulent kinetic energy are shown to maintain a similar shape, while decreasing in value with increasing Reynolds number. From the wall coordinate plots of velocity and temperature, both are shown to follow the logarithmic law of the wall. Near the center of the pipe at higher y^+, the expected deviation from law of the wall is seen.

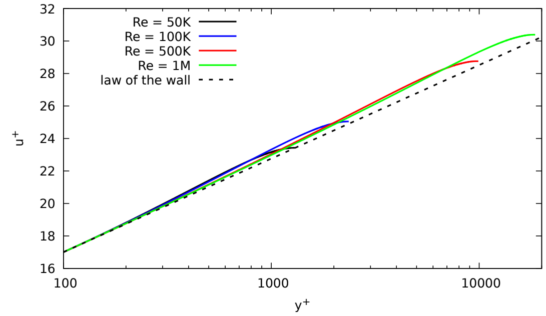
Comparisons between the predicted Darcy friction factors and Nusselt numbers are provided in Table 2.2. The observed agreement between the wall modeled approach and accepted correlations is quite good. For all cases, the percent difference between the wall modeled approach and correlations is less than 10%. It is expected that the wall modeled approach should become more accurate with increasing Reynolds number. This is reflected in the decreasing percent difference for both the friction factor and Nusselt number. Interestingly, the Nusselt number agrees the best at $Re =$ 500,000, although this is likely a mere coincidence.

Rod Bundle

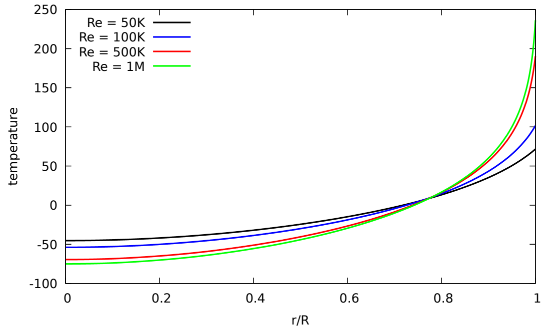
Wall functions have been demonstrated for a single rectangular subchannel geometry. The pins are on a square pitch, with a pitch-to-diameter ratio of 1.3263 and the case has a Reynolds number



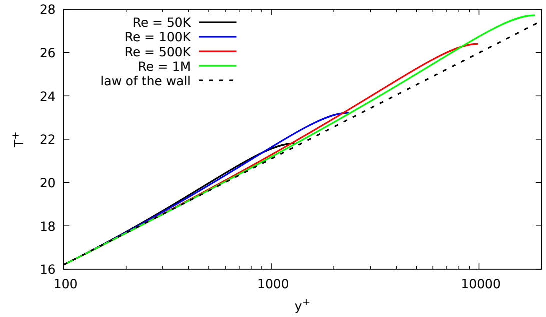
(a) Velocity (u/U)



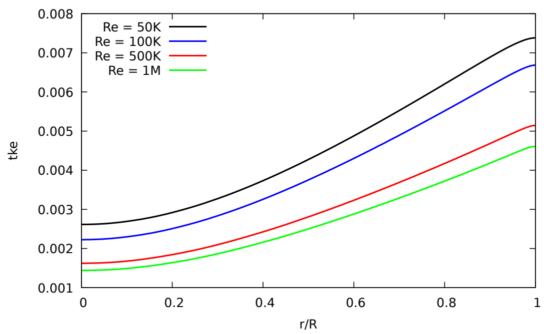
(b) u^+ (u/u_τ)



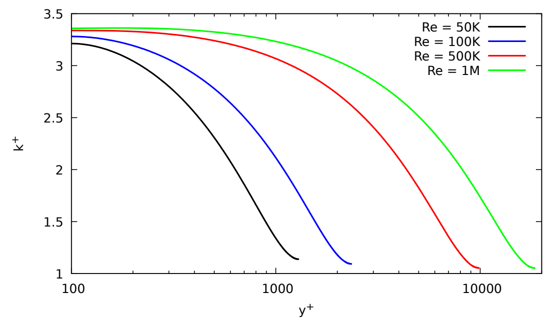
(c) Temperature $(T - T_b)\frac{q''}{\rho c_p U}$



(d) T^+ $(T_w - T)\frac{q''}{\rho c_p u_\tau}$



(e) Turbulent kinetic energy (k/U^2)



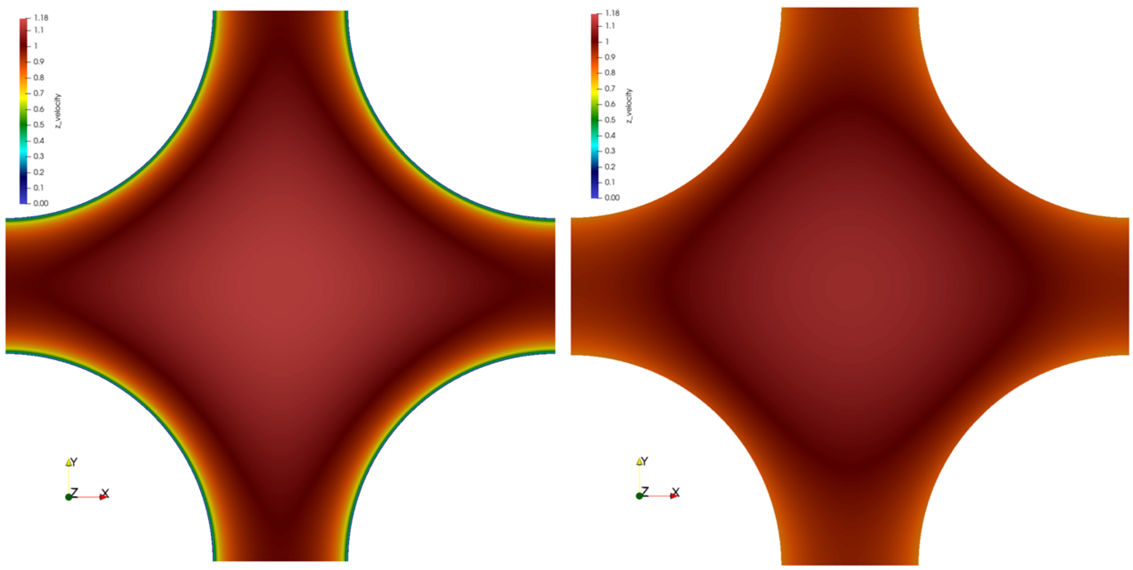
(f) k^+ (k/u_τ^2)

Figure 2.7: Velocity, temperature, and TKE profiles for the wall modeled approach at various Reynolds numbers for pipe flow

Table 2.2: Comparison of friction factors and Nusselt numbers to correlations

	friction factor			Nusselt number		
Reynolds	correlation	Nek5000	difference	correlation	Nek5000	difference
50K	0.0209	0.0190	9.9%	132.1	122.0	7.9%
100K	0.0180	0.0168	7.0%	230.0	217.1	5.8%
500K	0.0132	0.0127	3.9%	833.5	829.0	0.5%
1M	0.0117	0.0112	3.6%	1451.2	1483.4	2.2%

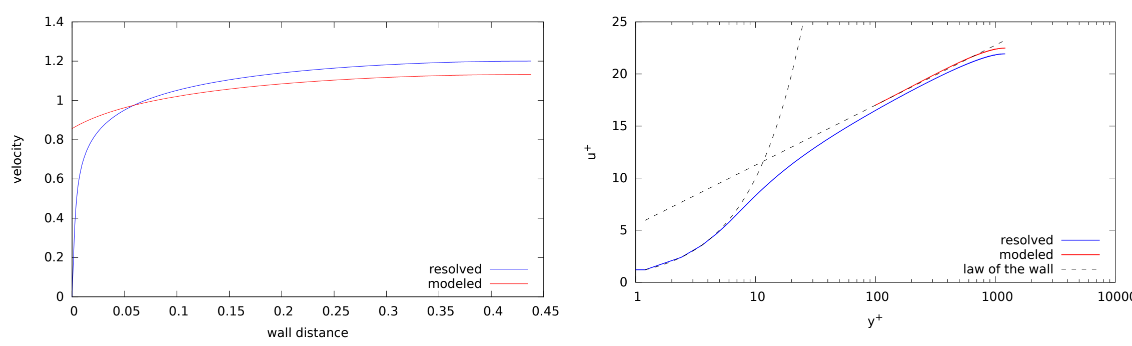
of 50,000 based on the pin diameter. Two cases are compared at fully developed conditions, one for the wall-resolved $k-\tau$ model and one for the $k-\tau$ model with wall functions. Colormaps showing the axial velocity distributions for the two cases are shown in Figure 2.8. The overall velocity profiles compare reasonably well. While some differences can be observed, these are mostly due to the slip boundary condition imposed with the wall function formulation in contrast to the no-slip condition for the wall resolved model. To further illustrate this, the velocity profile across the channel diagonal are shown in Figure 2.9. Profiles are provided in both standard and wall coordinates. Both cases show very good agreement in the logarithmic region to each other as well as to the law of the wall. Additionally, both models provide good predictions of the friction velocity with only an 8% difference between the two models.



(a) wall resolved

(b) wall modeled

Figure 2.8: Color map of the axial velocity predicted in the subchannel for (a) the wall-resolved $k-\tau$ model and (b) the $k-\tau$ model with wall functions.

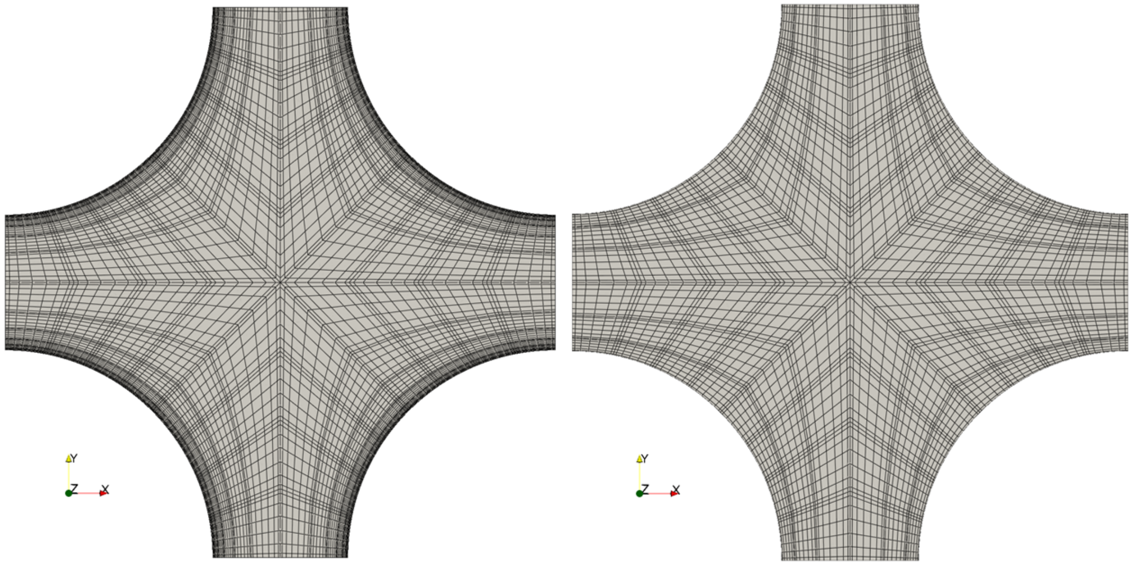


(a) velocity (u/U)

(b) u^+ (u/u_τ)

Figure 2.9: Comparison of velocity profiles between wall resolved and wall modeled RANS

A cross-section of the meshes used for each case are shown in Figure 2.10. While both cases have similar resolution in the bulk of the channel, it is apparent the wall modeled case requires considerably fewer elements near the wall. For this case, only half as many elements are needed, at higher Reynolds numbers the potential savings in computational cost can be even more significant.



(a) wall resolved mesh

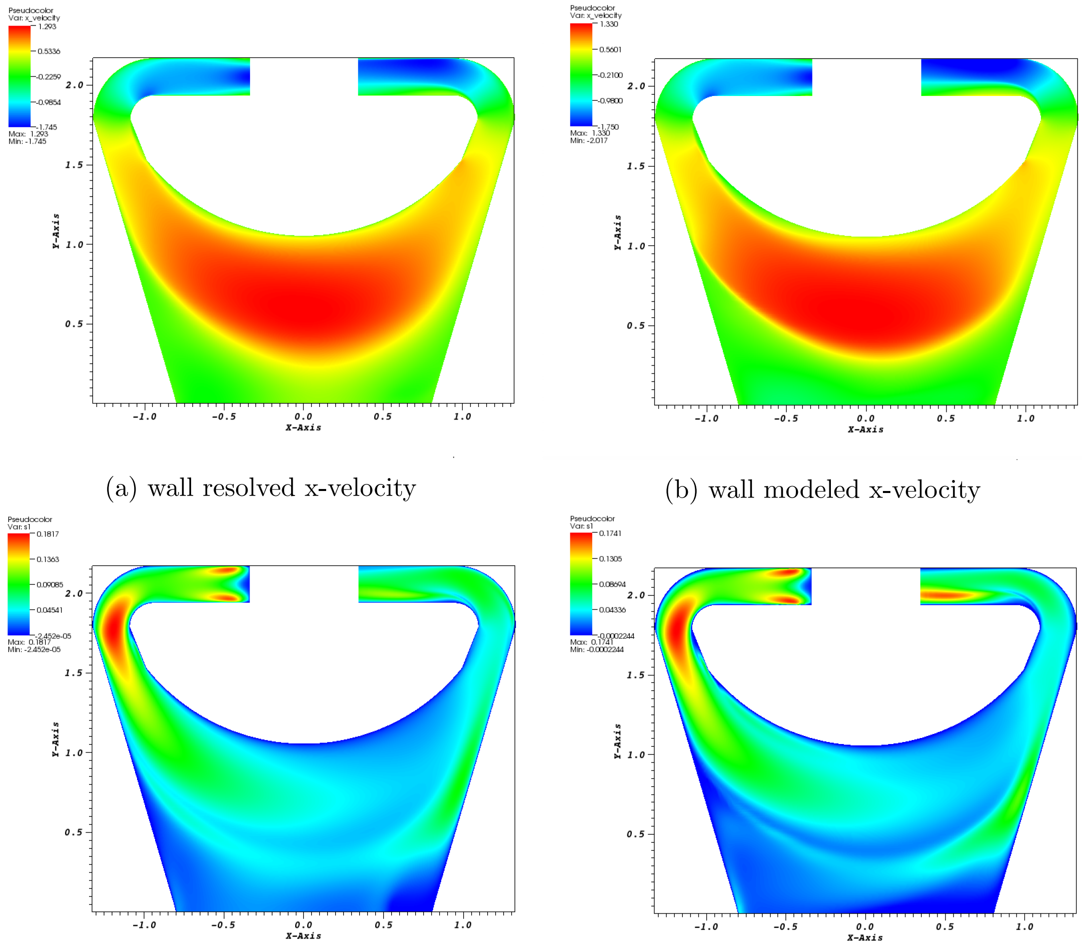
(b) wall modeled mesh

Figure 2.10: Cross section view of the computational meshes used for (a) the wall resolved case and (b) the wall modeled case

Molten Salt Fast Reactor Core

In addition to the cases presented above, we demonstrate the wall function implementation on . The MSFR core setup is representative of the proposed geometry for the Evaluation and Viability of Liquid Fuel Fast Reactor System (EVOL) concept developed by CNRS [22]. We performed RANS simulations of this design for a moderate Reynolds number of $Re = 40,000$ based on mean velocity through the core minimum diameter. This was done to facilitate the wall resolved model. The actual design calls for a significantly higher Reynolds number. The geometry is axially symmetric with x being the direction of the axis of symmetry. Simulations were performed using the wall resolved and the wall modeled $k - \tau$ model. Isocontours of the streamwise velocity and TKE at steady state are shown in Figure 2.11. As can be observed in these figures the wall modeled isocontours for both u and k are qualitatively as well as quantitatively close to the isocontours of the wall resolved case.

However, since this is a flow with large scale recirculation and wall modeled RANS is based on wall functions, which are derived using the law of the wall for attached flows. Thus, in this flow they are not expected to demonstrate full quantitative agreement with corresponding wall resolved simulations. This can be observed for example in Figure 2.12, showing profiles of streamwise velocity and TKE at various locations. Fig. 2.12 shows profiles across the inlet pipe at $x = -0.5$ and from $y = 1.935$ to $y = 2.17$. As can be observed, in this inlet part of the domain that the flow is still attached, the agreement between the wall resolved and wall modeled cases is very good for both



(a) wall resolved x-velocity

(b) wall modeled x-velocity

(c) wall resolved turbulent kinetic energy

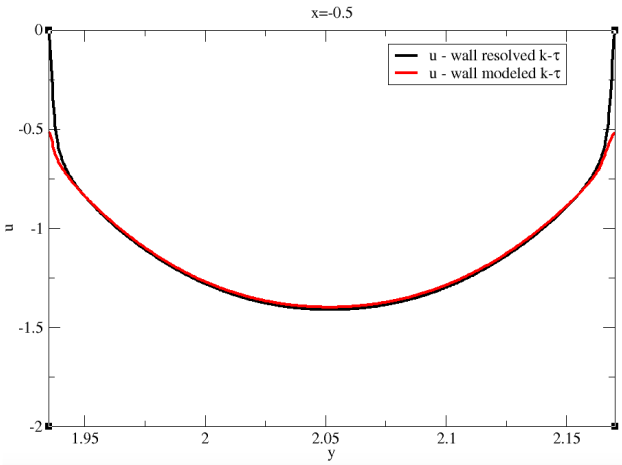
(d) wall modeled turbulent kinetic energy

Figure 2.11: Comparison of x-velocity (u/U) and turbulent kinetic energy (k/U^2) isocontours for the wall resolved and the wall modeled $k-\tau$ models.

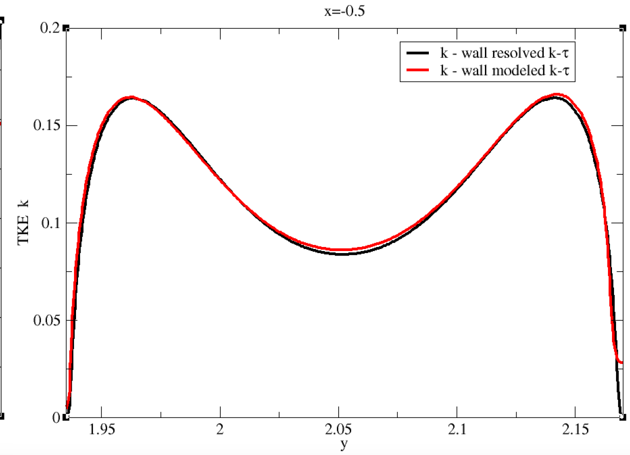
u and k. However, looking at $y = 0.5$ and from $x = -0.95$ to $x = 0.95$, which is well inside the large scale recirculation, agreement deteriorates but still maintaining the same qualitative behavior. Fig. 2.12 shows profiles across the outlet pipe at $x = 0.5$ and from $y = 1.935$ to $y = 2.17$ and as it can be observed, the streamwise velocity u and TKE k for the wall modeled case are over-predicted by more than 15% and 30%, respectively. Still, overall good qualitative agreement is observed in both the isocontours as well as profiles between the two cases.

3 NRC Support

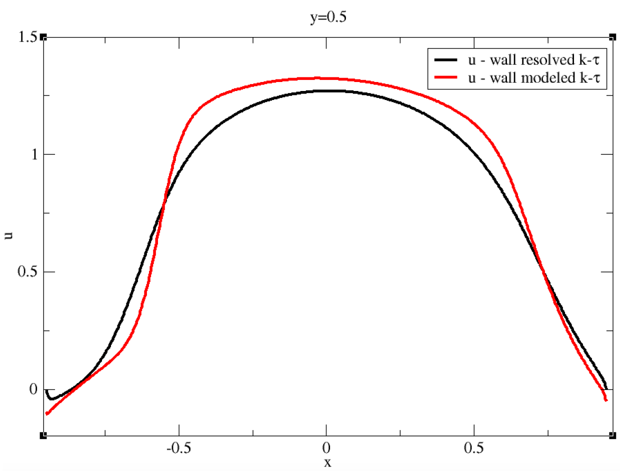
This year we have been continuing our support of and collaboration with the US Nuclear Regulatory Commission (NRC) staff members. They are working within the OECD/NEA Hymeres-2 program



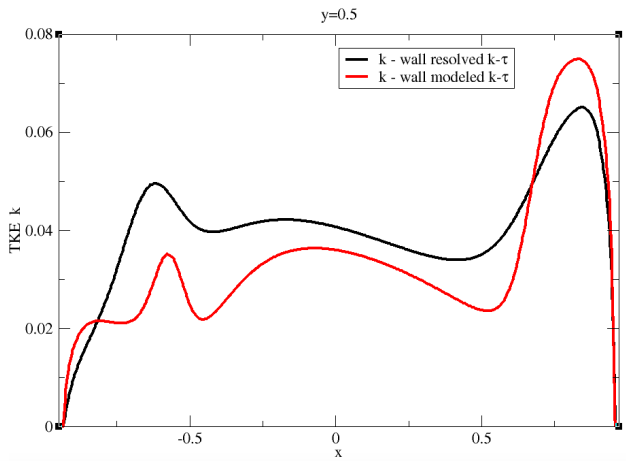
(a) x-velocity at $x = -0.5$ (inlet)



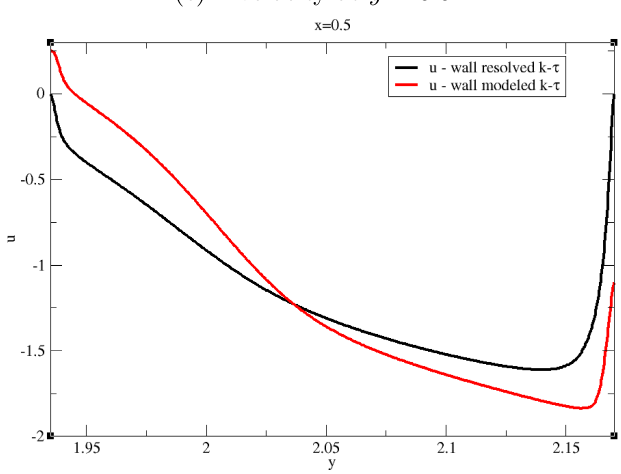
(b) turbulent kinetic energy at $x = -0.5$ (inlet)



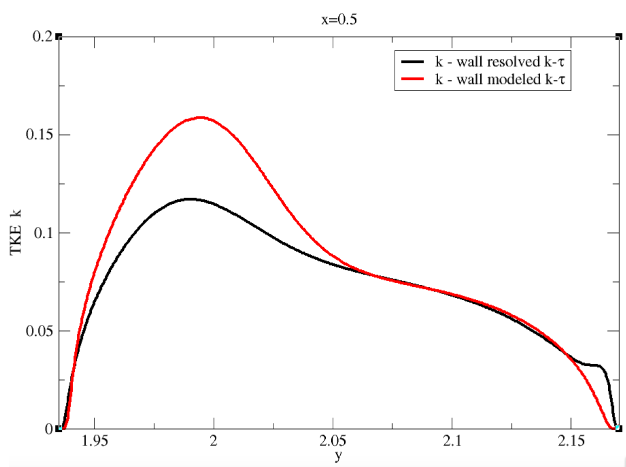
(c) x-velocity at $y = 0.5$



(d) turbulent kinetic energy at $y = 0.5$



(e) x-velocity at $x = 0.5$ (outlet)



(f) turbulent kinetic energy at $x = 0.5$ (outlet)

Figure 2.12: Comparison of profiles between the wall resolved and wall modeled $k - \tau$ models at various locations for x-velocity (u/U) and turbulent kinetic energy (k/U^2).

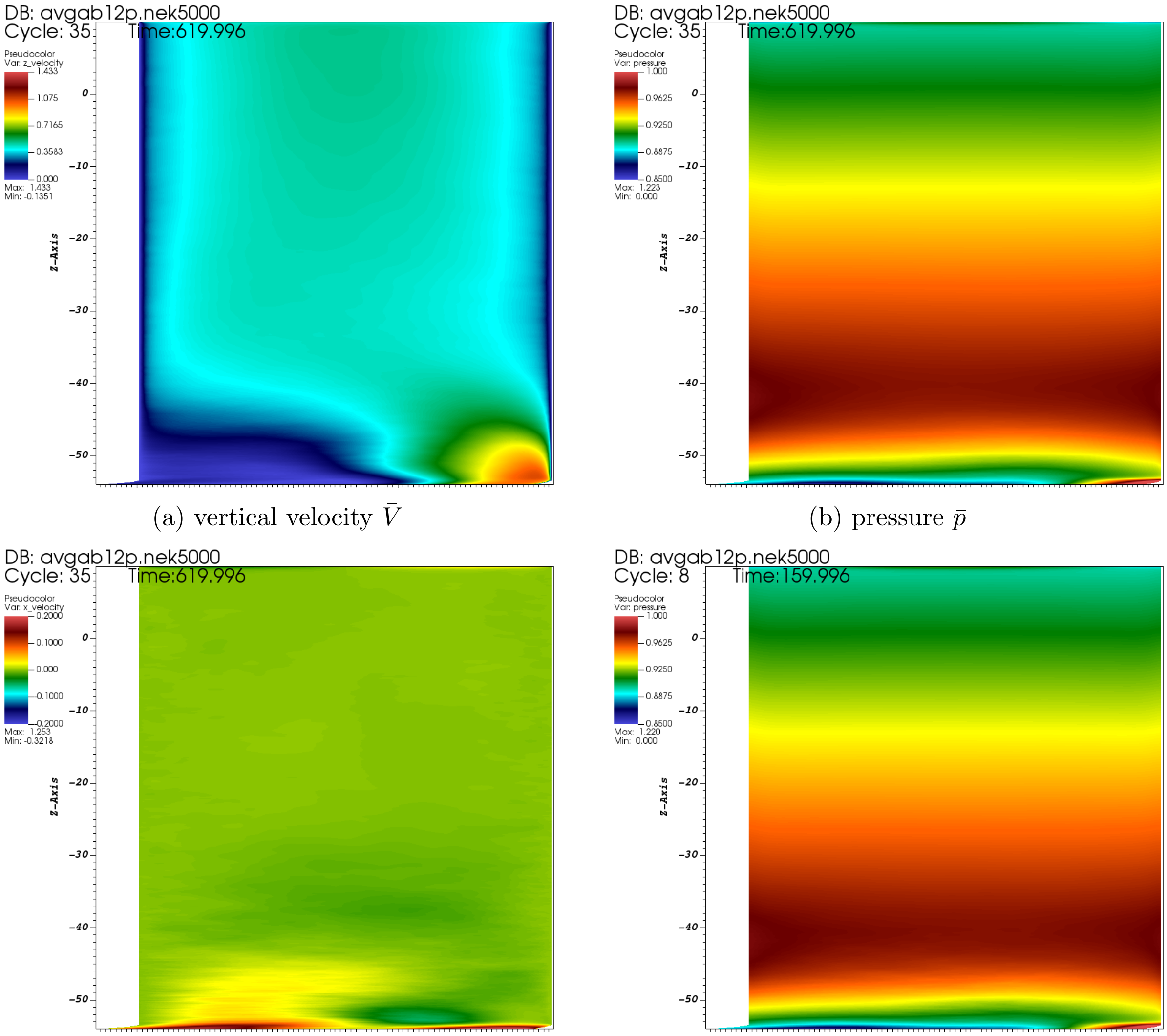
to improve the capabilities of computational dynamics (CFD) tools to model hydrogen mixing and mitigation strategies in nuclear power plant containments during severe-accident scenarios. Physical testing has been completed at the Paul Scherrer Institute (PSI) in Switzerland for the purpose of benchmarking the Nek5000 code which will provide details of the turbulent mixing phenomena in highly stratified containment environments.

The challenge problem has an enormously large ranges of transient turbulent scales which is typical for nuclear containment hydrogen mixing scenarios. This requires substantial spatiotemporal resolution of relatively high-speed jets within a domain that also includes significantly larger scale stratified layers and slow moving turbulent phenomena. The combination of the large domain, long transient initiation phase, a necessity to compute accurate-enough turbulent statistics with high fidelity in a quasi-steady phase, and the relatively small time-step requirements dictated by Courant-Friedrichs-Lewy (CFL) constraints in the jet region results in the need for significant computational resources for a large-eddy simulation (LES) campaign. The outcome of this cooperative effort between the NRC, OECD, and DOE/ANL will be presented at the final HYMERES-2 workshop later this year and will be a basis for future international benchmark and peer-reviewed publications.

In support of the three-dimensional severe accident safety analyses in nuclear power plants and its improvements, the U. S. NRC is involved with the OECD/NEA HYMERES-2 project which includes high fidelity testing of erosion processes in a layer of helium subject to flow from vertical jets and around obstacles. In a close collaboration of ANL with U. S. NRC staff, we have continued validation efforts using the PANDA facility data. The experiments performed at this facility include the 2014 OECD/NEA–PSI benchmark which concluded with the CFD For Nuclear Reactor Safety (CFD4NRS-5) workshop at ETH in Zurich. The latter benchmark is aimed at assessment of CFD code maturity and applicability to prediction of Fukushima accident events. These are mimicked in a gradual erosion of an initially stratified air-helium layer by a turbulent round jet consisting primarily of air. Mole fraction and temperature readings were taken at various points throughout the domain to record the erosion behavior, and mean and RMS velocity profiles were averaged over a long transient time. These data were the basis for comparison with CFD results from URANS and LES using Nek5000 and other codes [2, 3].

The turbulent jet erosion of a stratified air-helium layer acts as a surrogate problem used to validate post-Fukushima containment thermal hydraulics and gaseous mixing predictive models. This important problem for nuclear reactor safety is hindered by the challenges of a huge range of modeling scales, transition from forced to buoyancy driven flow with and without obstacles, and the turbulent mixing and erosion of a significantly stratified layer. In particular, the current focus of model improvement involves acquiring the validated reference solution of a stratified layer where the erosion processes like the ones observed in OECD/NEA PANDA benchmark and HYMERES-2 project deviate substantially from the common isotropic turbulence assumption used in lower-fidelity CFD turbulence models and in reduced order/dimensionality models.

This stage of the experiment used a single phase fluid while other variants of the tests would include a mixture of steam near saturation temperature with phase changes. The validation and development work will support the NRC's longer term goals related to the improvement of URANS based turbulence models in stratified layer erosion processes. This is an area where common isotropic turbulence models used in CFD codes have difficulty predicting the mixing behavior. The NRC has



(a) vertical velocity $\bar{V}$

(b) pressure $\bar{p}$

(c) x-velocity $\bar{V}_x$

(d) $\bar{p}$ over a shorter averaging interval

Figure 3.1: Time-averaged field slices in a single-bend configuration at $x = 0$.

worked with its OECD/NEA HYMERES-2 partners to get the testing approved and completed.

The ultimate goal of this work is to improve faster-turn-around lower-fidelity modeling of (anisotropic) turbulence- and buoyancy-driven mixing that are typically based on isotropic turbulence modeling closures.

3.1 Inlet Sensitivity Study

Despite having ("curved") axial symmetry, the HYMERES-2 inlet has a rather complicated shape consisting of multiple bends and changes of the pipe diameter. Due to the large disparity of spatiotemporal scales in the benchmark flow, every bit of simplification is important, including a possible simplification in the inlet geometry and its' turbulence level modeling. Naturally, the first question that must be addressed is whether more simple modeling, i.e. a fully-developed pipe profile, is a good enough approximation or if more of the upstream geometry and/or the inlet synthetic turbulence modeling is necessary.

Last year's preliminary LES results [4] put the 8% bound on maximum deviation of periodic inlet solution from more complicated geometric versions with various degree of details in upstream geometry and turbulence level. This year NRC staff in close consultations with ANL collaborators has made a thorough mesh convergence study of a stand-alone pipe problem and adopted the mesh improvements for both bend inlet and full model cases.

Figure 3.1 illustrates the findings with velocity components and pressure cuts in LES of a single bend configuration of HYMERES-2 turbulent inlet with improved meshing after extensive mesh convergence study of pipe flow at NRC. As noted in the previous report, all the details of geometry will be shown after a discussion with the experiment group. Also note that the view of the figures is focused on the upper portion of the straight vertical pipe including the pipe outlet (see Figure 3.4) that is the ultimate objective of this study due to its role in the inlet conditions for the full vessel geometry (see Section 3.2).

As expected the axial component (in the upper portion) and pressure distributions (Figures 3.1a–3.1b) reach statistical steady state quicker than other components (e.g. Figure 3.1c) with pressure being the fastest (cf. Figure 3.1d). The key feature of these simulations was one of the longest data collection time intervals used in this type of geometry with only a small fraction of them presented here as a part of debugging of Nek5000 file-average routines on ALCF systems.

Further analysis of the profiles by NRC stuff confirms that the fully-developed inlet (with recycling technique) is an accurate enough representation of the inflow boundary conditions for full geometry within the bound of 2%. On a side note, this thorough study also confirms that effects of the outflow boundary conditions are indeed confined within immediate vicinity of the outlet (e.g. Figure 3.1c).

In summary, this study has established that modeling the HYMERES-2 inlet as a short pipe with fully-developed turbulent flow using a recycling technique is adequate for modeling the full PANDA vessel entrance flow.

3.2 HYMERES-2 Turbulent-Inlet Quasi-Statistically-Steady Setup

After careful sensitivity LES of pipe inlet geometry at conditions of the benchmark mentioned in Section 3.1, we have implemented and tested the LES solutions on a new mesh that is in production runs now. This mesh has a slightly different and more-efficiently-clustered element layout that takes advantage of the earlier study settling the difference of the inlet flow modeling between the straight long and bend short pipe cases both involving recycling/fully-developed flow. As a first step, we focus on the HYMERES-2 isothermal configuration without a disk obstacle. The hydro LES is not only somewhat faster and easier to obtain but also as accurate as the full thermal case with

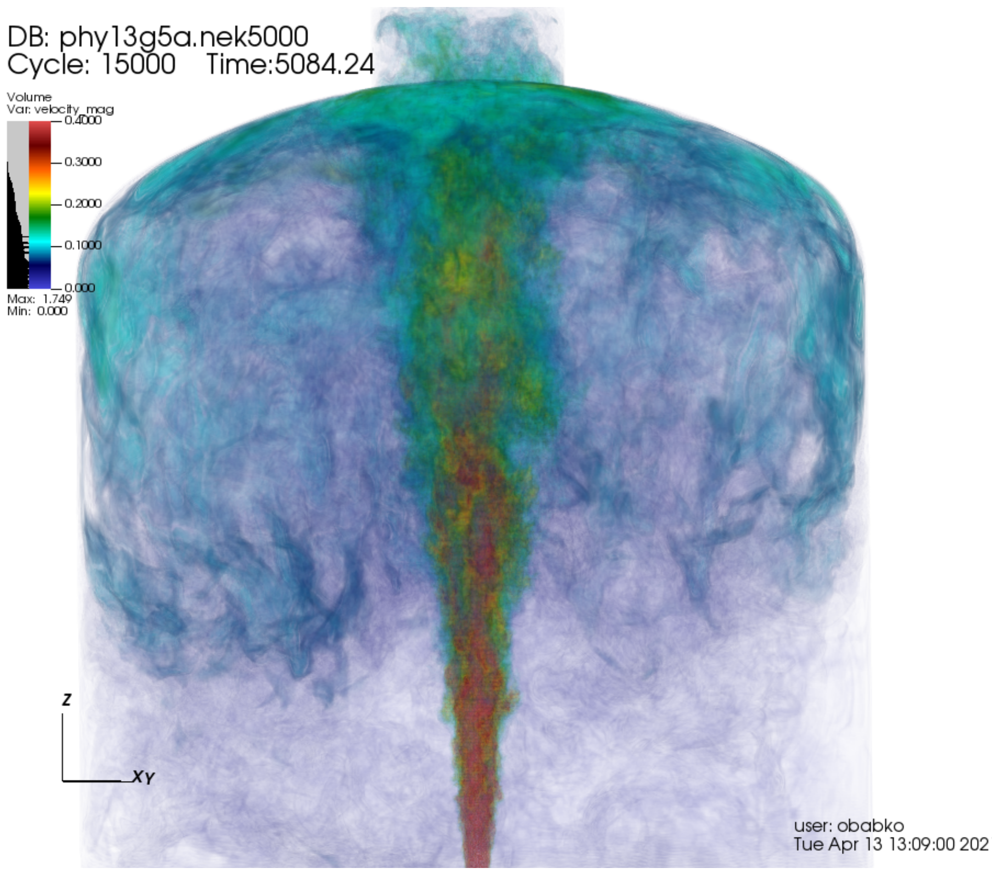


Figure 3.2: Instantaneous vertical velocity in HYMERES-2 quasi-statistically steady setup .

buoyancy and mass transfer in the long-term flow evolution due to ultimate erosion of helium layer that correspond to one of the regimes of data acquisition in the PSI experiment.

The volume rendering of velocity magnitude in Figure 3.2 illustrates the long term hydro flow evolution for the final mesh and specification of cross-verification and validation campaign that also will be compared further heat and mass transfer runs with buoyancy effects. This figure shows initial low-resolution runs and higher polynomial degree case has been also computed at INL on Sawtooth that has remarkable (now days) week-long-run queues.

The next steps are to finish these higher-resolution production runs and in parallel finish the setup and run the heat and mass transfer case with buoyancy validating the results against the PSI's HYMERES-2 data.

3.3 Miscellaneous and Future work

In addition to the primary focus of HYMERES-2 LES setup and runs, we have been also involved with additional training on ALCF's workshops and submitted the ALCC, INCITE, and ALCF's Director Discretionary allocations. We found both ALCF Computational Performance and Simulation,

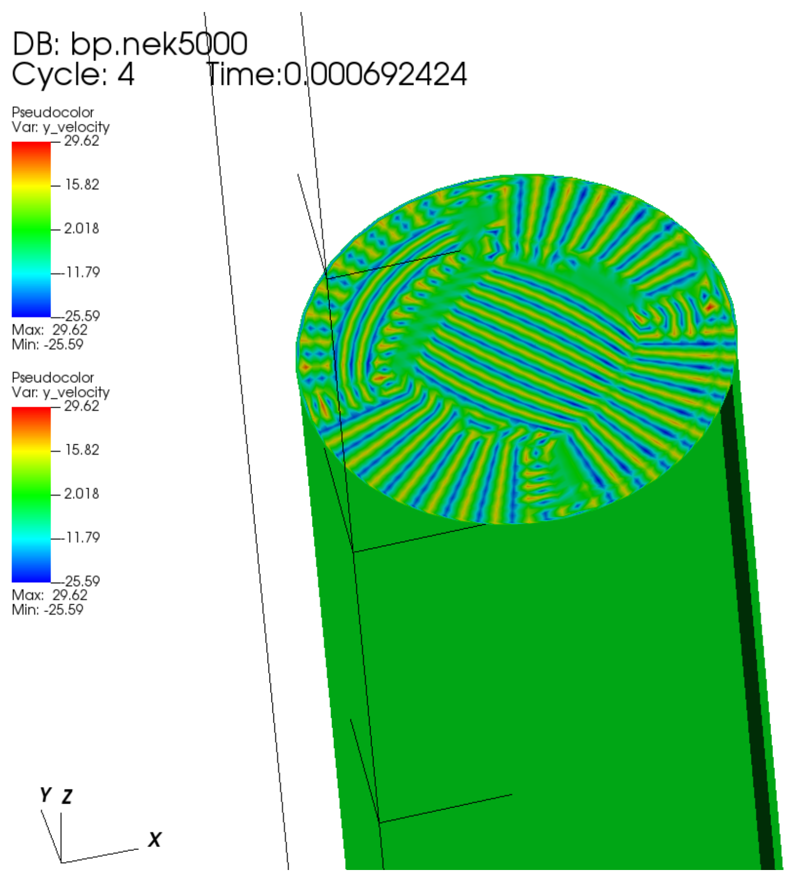


Figure 3.3: Bend setup's outlet with small initial time-step.

Data, and Learning workshops to be excellent venues for quick setup, debugging and overnight longer-flow-evolution testing at scale including pre- and post-processing setups with all ingredients being crucial for a successful start-up and modification of an LES simulation campaigns. In one of

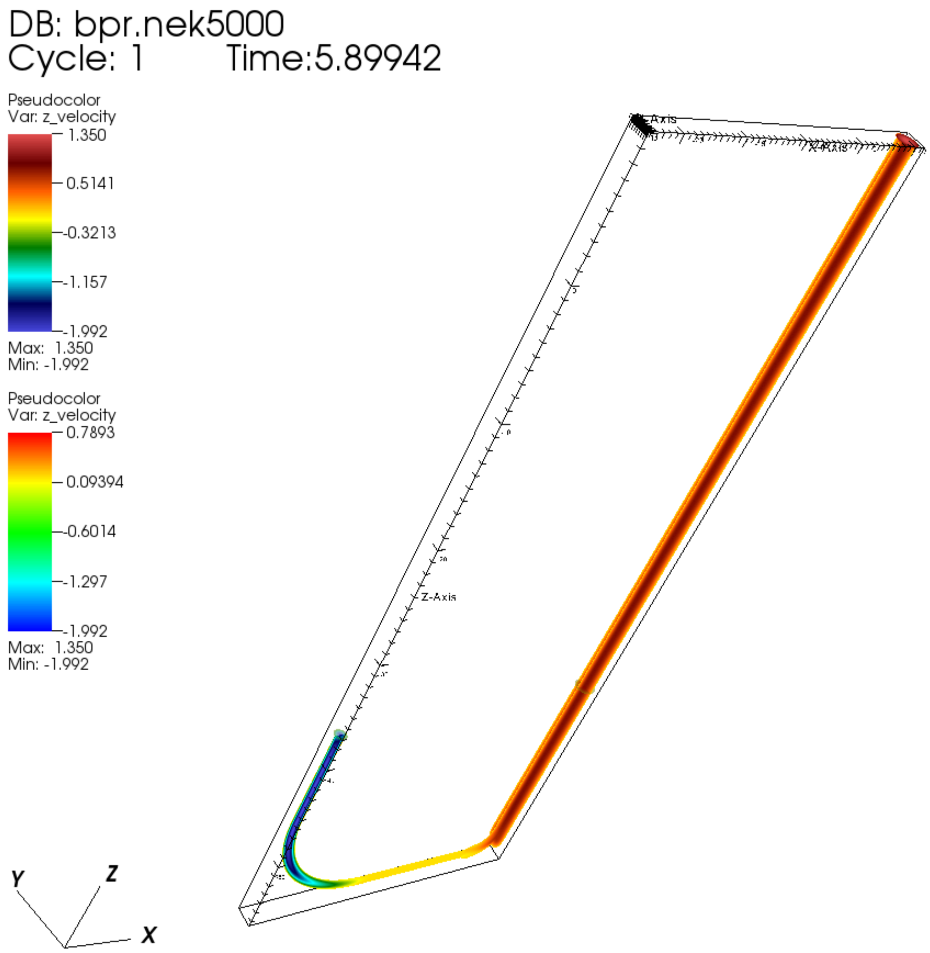


Figure 3.4: Fixed bend setup case.

this workshop we have successfully troubleshooted another bend case setup whose flow solution has been failing for multiple reasons including initial pressure converge issues and runaway oscillations at the pipe outlet. Figure 3.3 shows y-component of velocity that is a better illustration of the problem encountered. Using the workshop's fast priority/dedicated queues we were able quickly to locate most of the culprits and found a fix (Figure 3.4) involving higher initial time-step applied together with the operator-integration-factor splitting (OIFS) time-stepping/extrapolation scheme [23].

In summary, this primary year's focus of the ANL-NRC collaboration on benchmark tests resulted in even more efficient geometry and inlet modeling simplification after a careful study of the inlet sensitivity solutions. This modeling choice significantly simplifies the next step of the cross-V&V HYMERES-2 project where the higher resolution of isothermal long-term flow evolution and heat and mass transfer solution will be obtained and compared.

4 Quadratic Tet-to-Hex

In the design of advanced nuclear energy technologies , components are adopting complicated designs, such as fuel assemblies with spacer grids [24], [25], helical coil steam generators [26], [27], random pebble beds [28], printed circuit heat exchangers [29], etc. For Computational Fluid Dynamics (CFD) simulations, it is essential to generate quality meshes for these complex designs.

The DOE NEAMS program aims at developing advanced modeling and simulation tools and capabilities to accelerate the deployment of advanced nuclear energy technologies, including light-water reactors (LWRs), non-light-water reactors (non-LWRs), and advanced fuels. Nek5000, developed under the NEAMS program, is an open-source CFD code based on the Spectral Element Method (SEM) [30]. It has shown great scalability from meshes with as few as a couple thousand elements, to tens of millions of elements (billions of degrees of Freedom) [31] . The spectral Element Method is quite different from the Finite Volume Method (FVM). CFD codes using FVM usually adopt a variety of meshes type, such as tetrahedral, hexahedral, wedge, pyramid, polyhedron, which brings great flexibility to the meshing strategy.

In SEM, variables are described as a piecewise polynomial expansion. The foundational idea is to minimize the error over a chosen space of piecewise polynomials. Nek5000 uses the Gauss-Lobatto-Legendre (GLL) polynomials to represent the variables like velocity, pressure, and temperature. The SEM converges exponentially in N (polynomial order), which implies that significantly fewer grid points per wavelength are required to accurately propagate a signal over the extend times associated with flow simulations at high Reynolds number. However, Nek5000 only accepts hexahedral elements. This requirement makes developing a mesh for complicated problems quite challenging.

Nek5000 decomposes the computational domain into quadratic hexahedral elements. Traditionally, a block-method is used to generate such a mesh: the domain is subdivided into smaller blocks. The union of the blocks corresponds to the full domain. Each block is then divided into conformal hexahedral elements. This method can be used for geometries with some complexity, however if the geometric complexity reaches a certain degree, using the block method becomes time consuming and sometimes impossible.

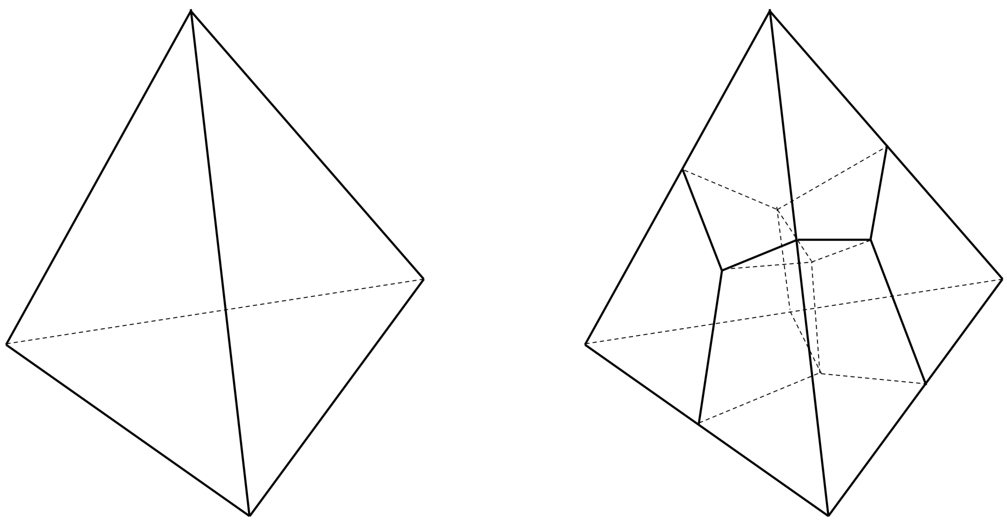
To address these issues, we proposed a tet-to-hex meshing method in our previous paper [25]. The tet-to-hex meshing method can utilize the high flexibility of a pure tetrahedral mesh to conform to the geometry, but it can still maintain the higher-order accuracy of Nek5000. Moreover, in complex geometries, it can do so at a computational cost and accuracy comparable to the block-structured mesh when available. However, in our previous paper, the tet-to-hex conversion is linear. In this paper, we improved this approach to its quadratic version.

4.1 Strategy for pure hexahedral mesh for complicated domains

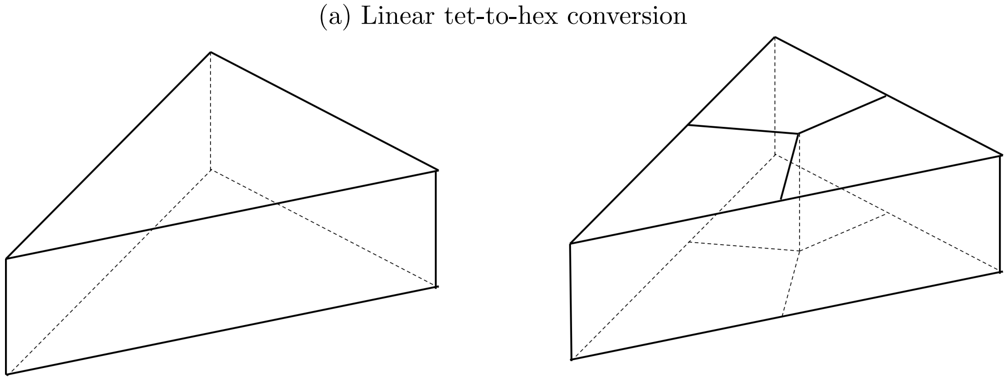
Significant effort has been invested by other investigators to develop an automatic pure hexahedral meshing algorithm [32]. However, the progress is still limited. A lot of human interference with the meshing procedure is still needed to make a valid pure hexahedral mesh. Moreover, these algorithms usually do not consider boundary layers, which are essential for CFD simulations.

Tet-to-hex mesh conversion

In our previous work [25], we have developed the linear tet-to-hex approach to mesh complicated geometries. First, a pure tetrahedral mesh is generated. Then each tetrahedral element is divided into 4 hexahedral elements as shown in Figure 4.1a. The boundary layers are constructed with wedge elements, and each wedge element is divided into 3 hexahedral elements as shown in Figure 4.1b. At this point this conversion is linear, which means no curvature exists for element edges. To ensure boundary curvature is captured, we have to perform an extra step to project the linear mesh to the boundaries, which is discussed in the next subsection.



(a) Linear tet-to-hex conversion



(b) Linear wedge-to-hex conversion

Figure 4.1: First order element conversions

Mesh morphing projection based on GLL points

After a linear hexahedral mesh is created, we must project the mesh to the geometry boundaries. This step is necessary to ensure higher-order accuracy. Two changes are made to the mesh. The first change is to project the new linear hexahedral elements (hex8) to conform with the curvature.

The converted hex8 elements are not conformal by default. This is because only the vertices of the tetrahedral elements (tet4) are conformal to the geometry. This projection step will enforce the conformality of the hex8 mesh. The second change is to convert the linear hexahedra elements (hex8) to quadratic hexahedral elements (hex20). The added mid-edge points are then also projected to the geometry, forcing the elements to be conformal to the domain again. The difference between hex8 and hex20 elements is shown in Figure 4.2. In a hex20 element, mid-edge points (marked in red in Figure 4.2) are added to conform to the geometry.

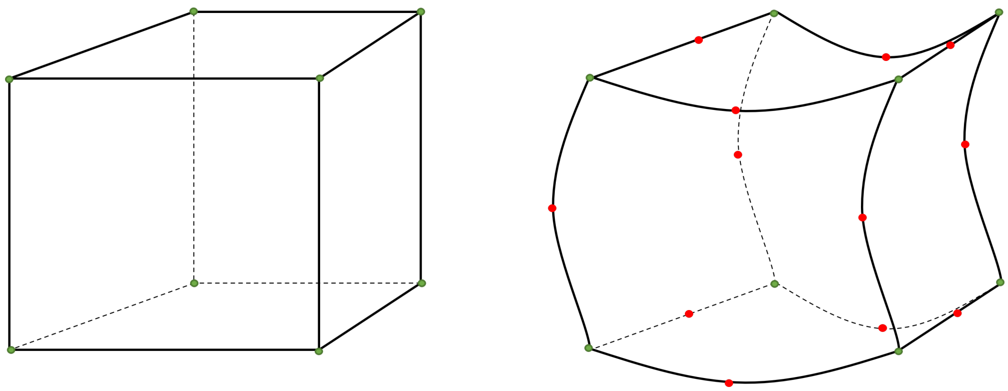


Figure 4.2: A hex8 element (left) and a hex20 element (right)

In Nek5000, we used the Laplacian Equation to solve the displacement of the mesh, with a user provided boundary displacement vector as a boundary condition. As a reminder, the displacement of the mesh does not only happen at boundaries, it also propagates to the internal mesh. This preserves mesh quality and boundary layers. This approach is similar to that used by the mesh smoother [33]. Figure 4.3 shows the meshes before and after morphing. However, this approach requires users have a certain familiarity of the code. Additionally, because there is no CAD engine coupled to Nek5000, assigning mesh projection vectors for over-complicated surfaces becomes impossible. The current compromise is to only project surfaces that are most important to the problem, which is a common strategy to balance computational power and resolution.

Transfinite projection based on quadratic mesh

Commercial and open-source meshing codes (ANSYS-meshing [34], Gmsh [35], etc) could directly generate quadratic meshes that are conformal to the computational domain. Similar to our previous work, we focus on two types of elements: tetrahedral and wedge elements. Their quadratic versions are shown in Figure 4.4. The tet10 element is the quadratic version of tet4 element, with mid-edge points to describe curvature. This also applies to wedge15 elements. Here we follow the definition of the Exodus mesh [36] format, as the current mesh converter uses an Exodus mesh.

Transfinite interpolation or mapping has been widely used to in meshing and post-processing [37]-[38]. The basic idea is to reconstruct the element faces based on the curvature of the edges. Specific to our application, we need to do two types of transfinite interpolation: one is for quads and the other for triangles.

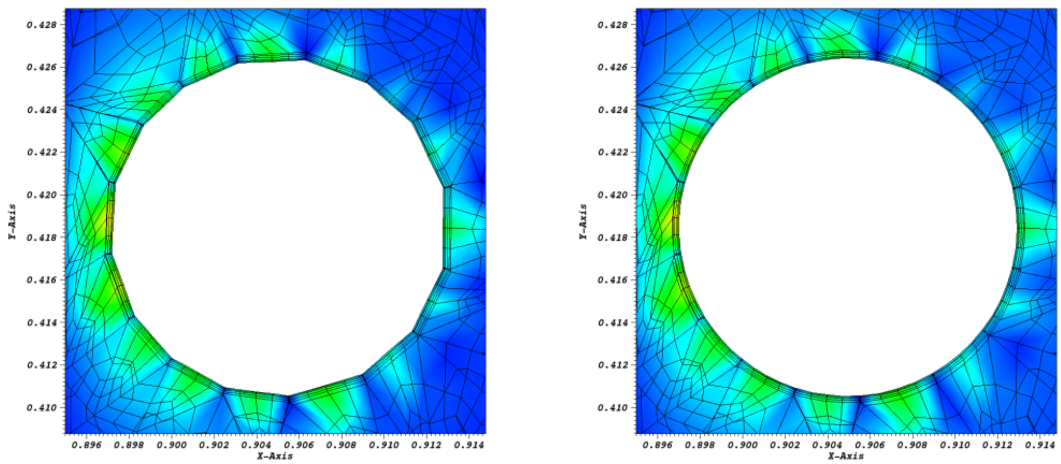


Figure 4.3: An example of mesh morphing showing (left) the linear mesh before morphing and (right) the quadratic mesh after morphing – the color shows the displacement magnitude

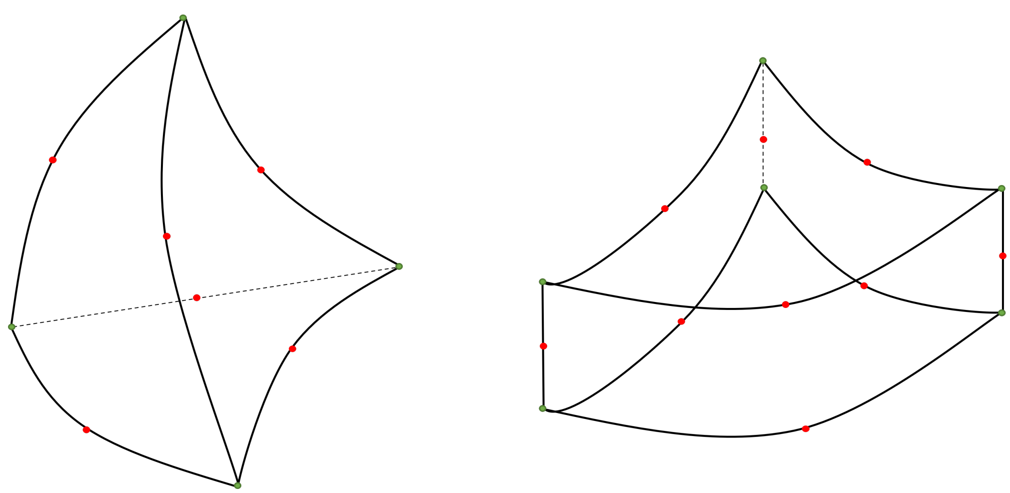


Figure 4.4: Quadratic elements, (left) tetrahedron (tet10) and (right) wedge (wedge15)

For the quad transfinite interpolation [39], assuming the quadratic quad is described by four curves; $c_1(u)$ and $c_3(u)$ describe one pair of edges on the opposite side while $c_2(v)$ and $c_4(v)$ describe the other pair. Then any point (u, v) on this quad can be calculated using the following equation:

$$\begin{aligned}S(u,v) = {} & (1-v)\cdot c_1(u) + v\cdot c_3(u) + (1-u)\cdot c_2(v) + u\cdot c_4(v) \\ & - [(1-u)(1-v)O_{1,2} + uvP_{3,4} + u(1-v)P_{1,4} + (1-u)vP_{2,3}]\end{aligned} \tag{28}$$

Where, e.g., $P_{1,2}$ is the intersection of curve $c_1(u)$ and $c_2(v)$, and the ranges of u and v are $(0 \leq u \leq 1)$ and $(0 \leq v \leq 1)$ respectively.

For the triangle transfinite interpolation ([40]), the equation is more convoluted. Inside a triangle, any points can be projected to two edges in the parallel direction of the other edge. This generates three coordinates to describe this point, λ_1, λ_2, and λ_3. However $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Assuming edges of the triangle can be described by the function $\hat{v}(\lambda_1, \lambda_2, \lambda_3)$, then any point inside this triangle can be determined by the following function.

$$\begin{aligned}S(\lambda_1,\lambda_2,\lambda_3) = {} & \lambda_1\left[\hat{v}(1-\lambda_2,\lambda_2,0) + \hat{v}(1-\lambda_3,0,\lambda_3) - \hat{v}(1,0,0)\right] \\ & +\lambda_2\left[\hat{v}(0,1-\lambda_3,\lambda_3) + \hat{v}(\lambda_1,1-\lambda_1,0) - \hat{v}(0,1,0)\right] \\ & +\lambda_3\left[\hat{v}(\lambda_1,0,1-\lambda_1) + \hat{v}(0,\lambda_2,1-\lambda_2) - \hat{v}(0,0,1)\right]\end{aligned} \tag{29}$$

Using the equation presented above, we can perform the quadratic tet-to-hex and wedge-to-hex conversion as shown in Figures 4.5 & 4.6. While other methods exist to reconstruct the surfaces, Nek5000 already uses transfinite interpolation to construct the GLL points on the faces of hex20 elements. This makes it a convenient method to use for this application. In this work, we utilize the equations above to divide quadratic tetrahedral (tet10) and wedge (wedge15) elements into quadratic hexahedral (hex20) elements. Through this approach, the final hexahedral mesh will be conformal to the geometry curvature to 2nd order accuracy, with no need to morph the mesh in Nek5000. This feature is implemented into an experimental branch of *exo2nek*, one of the official mesh converters provided with Nek5000.

4.2 ROCOM experiment

In this part, we will present one example case that has benefited from the quadratic tet-to-hex meshing approach. The ROCOM experiment [41], [42] is a test facility for the investigation of coolant mixing in the primary circuit of PWRs. It reproduces the primary loop of a German KONVOI-type reactor. The ROCOM facility was built to provide high resolution data for CFD code validation.

In this work, we focus on the downcomer part of the reactor vessel as shown in Figure 4.7. The flow enters the downcomer from the four inlet pipes on the top, and then exist the domain after it

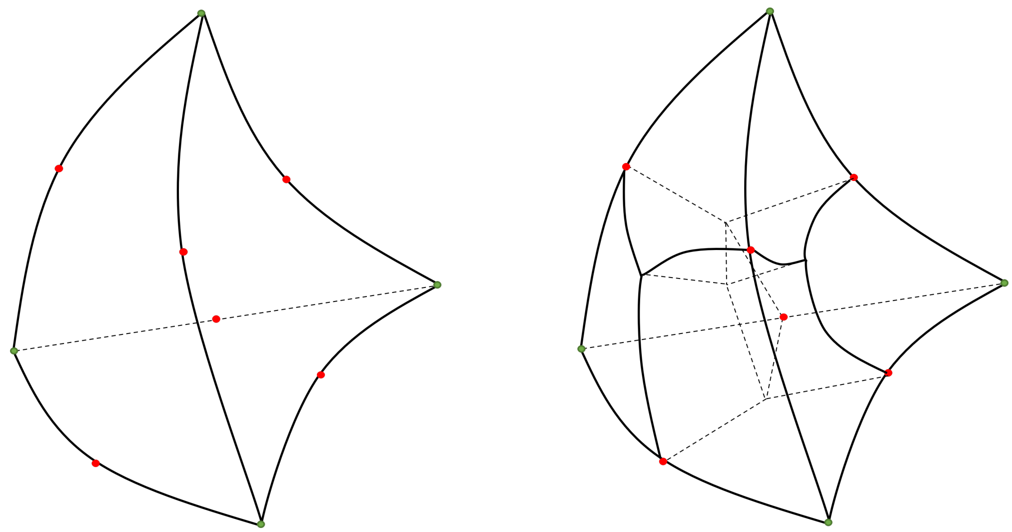


Figure 4.5: Quadratic tet-to-hex

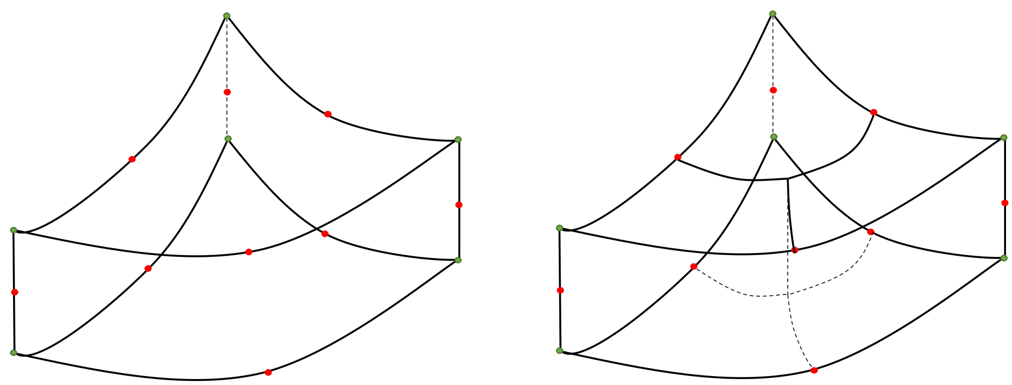


Figure 4.6: Quadratic wedge-to-hex

flows through the equalizer. The computational domain excluded the reactor core in this research. The mesh was first generated in ANSYS-meshing with quadratic tetrahedral and wedge elements. Then through the quadratic tet-to-hex conversion, we can obtain the quadratic hexahedral mesh in Nek5000. The mesh in ANSYS-meshing has approximately 0.5M tet10 elements and 75,000 wedge elements. The mesh in ANSYS-meshing is shown in Figure 4.8.

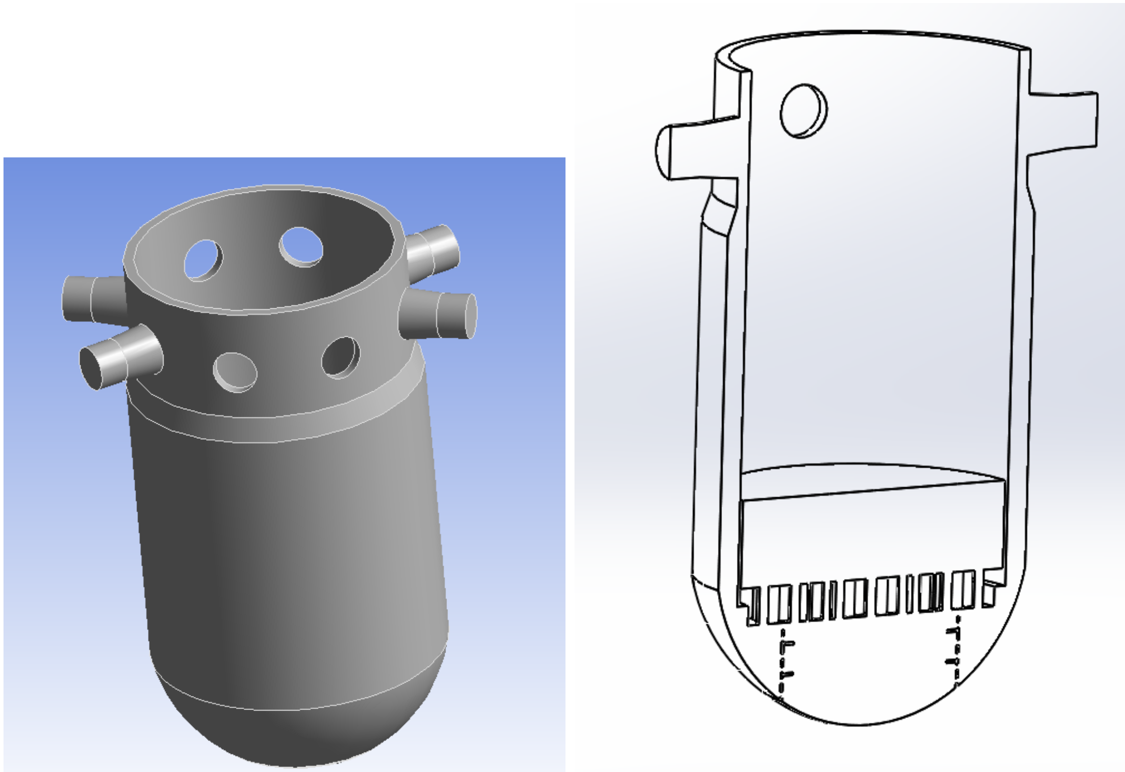


Figure 4.7: The ROCOM reactor vessel downcomer showing the (left) whole view and (right) sliced view

After conversion to a pure hexahedral mesh, there are around 2.5 million elements in total. With polynomial order of 5, the number of degrees of freedom is 312 million. The converted mesh in Nek5000 is shown in Figure 4.9. In this case, we set Reynolds number of 5000, based on vessel diameter and inlet velocity. The flow field of this demo case is presented in Figure 4.10 shortly after initialization. The number of pressure iterations drops to 20-30 after the initialization stage, which is a reasonable number. However, we will avoid diving into the detail of this case, as it is not the purpose of this work.

Here, we presented the development of a quadratic tet-to-hex conversion to generate pure hexahedral mesh. This process utilizes the robust tetrahedral meshing algorithm available in commercial and open-source meshing codes. A direct conversion of quadratic meshes preserves the curvature of the computational domain. It also saves the user an extra step of mesh morphing.

This technique does not solve all the problems in meshing due to its highly case and problem sensitive nature. For example, a tet-to-hex mesh will require more elements to describe the

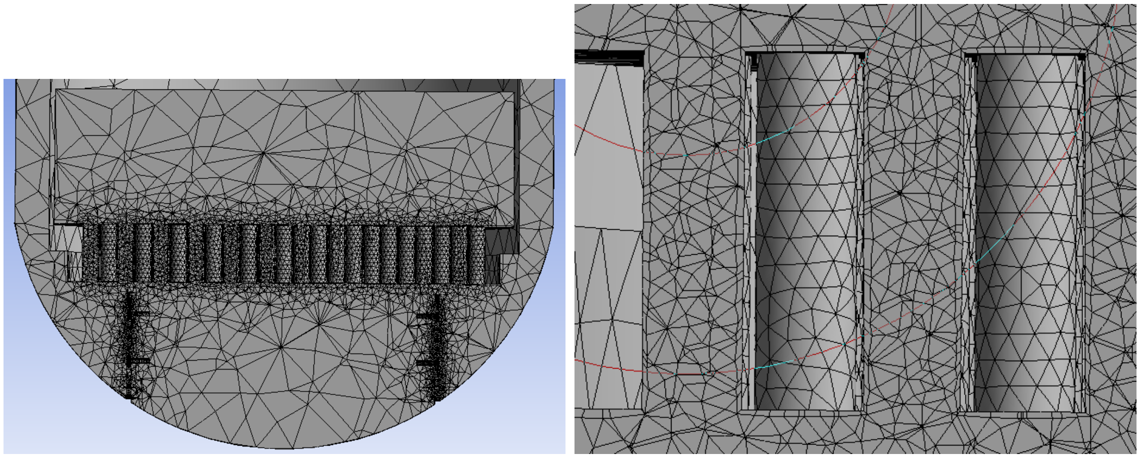


Figure 4.8: Mesh of the ROCOM facility in ANSYS-meshing using quadratic tetrahedral and wedge elements

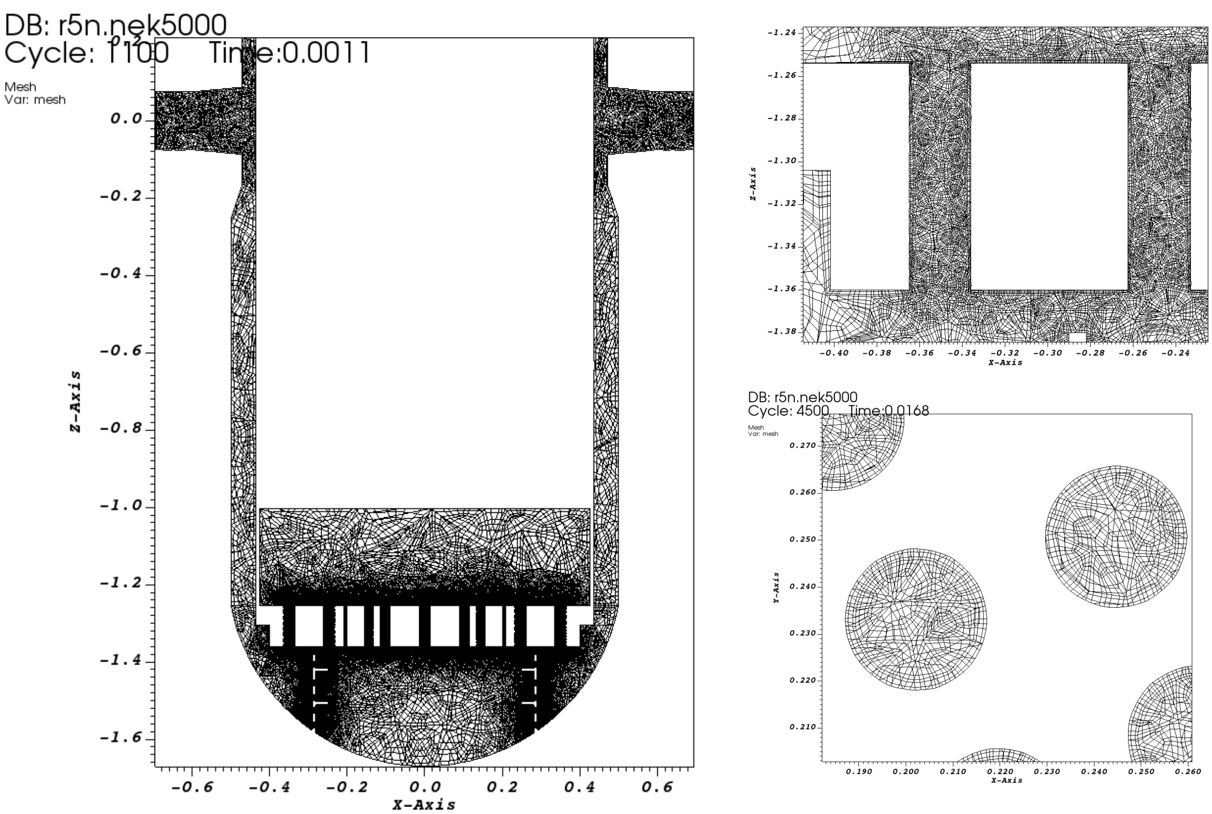


Figure 4.9: Mesh of the ROCOM facility in Nek5000 with quadratic hexahedral elements

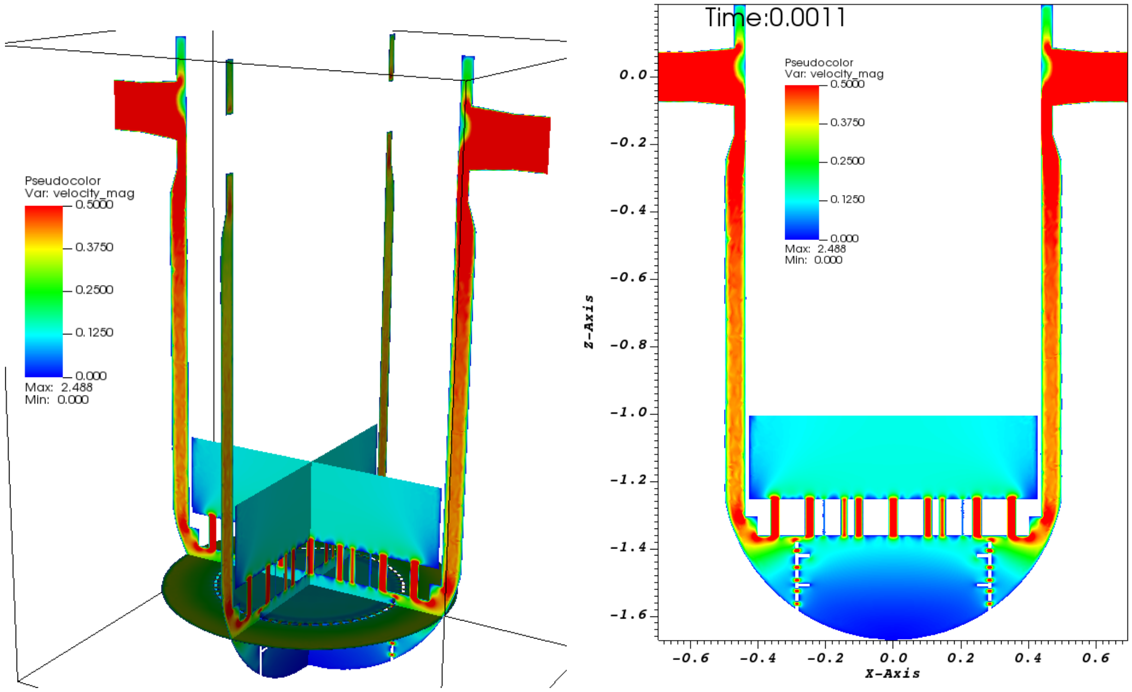


Figure 4.10: Flow field of the ROCOM facility in Nek5000

geometry compared to a blocked mesh, which also challenges users' available computational power. Additionally, the orientation of tet-to-hex elements can sometimes restrict the max time-step in the simulation. However, it does represent a very robust method of producing high-quality meshes for complex geometries. With the recent development of NekRS, the GPU variant of Nek5000, and the current push towards Exascale computing, larger and larger element counts can be achieved. Several GPUs could reach an equivalent computational power of hundreds of CPUs, which helps to ease the limitations of tet-to-hex approach.

5 Tutorials and Training Activities

As part of the Nek5000 documentation [43] several tutorial cases are provided. This year, to better serve the needs of users, we have begun revising the existing tutorials, added a new basic instructional tutorial, and hosted a virtual training seminar. The existing tutorials are being revised for better readability and usefulness to the user. Based on user feedback, they currently do a reasonable job of instructing users on what to do to perform a simulation, however they generally do not offer enough description on why something is done. This allows a user to perform that specific simulation, but does not offer any insight into how to apply the described capabilities to their own, usually unique, cases. Additionally a new tutorial was added that covers one of the most basic aspects of using Nek5000, setting inlet boundary conditions, addressing a significant gap in the current instructional offerings. Finally, a virtual training session was held for participants from Idaho National Laboratory and Argonne National Laboratory that leveraged the problem setup from this new tutorial.

5.1 Updates to tutorials

Conjugate heat transfer

The first tutorial that has been revised is the conjugate heat transfer tutorial. This was spurred by updated user input which ties the problem setup to a set of high-fidelity simulations available in literature [44]. By tying the simplified problem available in the tutorial to a larger scale simulation allows users to envision a path from the tutorial cases to the more complex cases they seek to simulate.

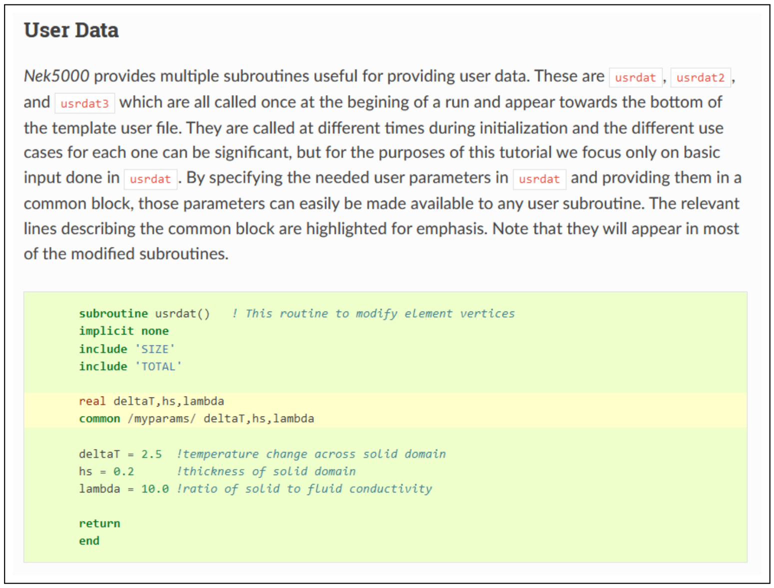
This tutorial guides the user through all the steps of the case setup, including the generation of a conjugate heat transfer mesh using *genbox* and *preNek*. The previous version of this tutorial generated the mesh in three sections, one for the fluid domain and two for the solid domain. As part of the update it has been simplified to one each for the fluid and solid domains. This makes the process easier to follow and simplifies merging the meshes using *preNek* into a single step.

The content and formatting of the tutorial documentation has also been revised for greater clarity. When the user is asked to modify parameters from the standard template files, exposition is provided as to what the modification is doing and why it is necessary. For example, the section explaining how to provide user input data for use in the `.usr` file describes the standard practice of declaring and using a common block so the data can be made available throughout the `.usr` file. Highlighting has also been added to the included `.usr` file subroutine examples to illustrate these changes, as can be seen in Figure 5.1.

Finally, in addition to the above updates, the user is provided with a complete set of case files that can be used in the event that they are unable to follow and execute the necessary steps described in the tutorial. This helps to ensure that the user is able to at least run the case.

Fully developed laminar flow

A completely new tutorial has been added to NekDoc. This tutorial is envisioned as one of the simplest possible cases that could be run and is intended as a problem a beginning user should be familiar with. As such, dimensional values are provided for all of the input parameters. The simulated case is for fully developed laminar flow in a channel with a constant wall heat flux applied. This is depicted in Figure 5.2. This case was chosen as it covers the most basic flow scenario that users may be interested in, a simple heated flow with a defined inlet and an outlet. The other provided tutorials are all periodic cases and the fully developed laminar flow case guides users through prescribing given profiles as Dirichlet inlet conditions. Additionally, this case has analytic solutions that can be compared against to verify that the case has been setup and run correctly.



User Data

Nek5000 provides multiple subroutines useful for providing user data. These are `usrdat`, `usrdat2`, and `usrdat3` which are all called once at the begining of a run and appear towards the bottom of the template user file. They are called at different times during initialization and the different use cases for each one can be significant, but for the purposes of this tutorial we focus only on basic input done in `usrdat`. By specifying the needed user parameters in `usrdat` and providing them in a common block, those parameters can easily be made available to any user subroutine. The relevant lines describing the common block are highlighted for emphasis. Note that they will appear in most of the modified subroutines.

```
      subroutine usrdat()   ! This routine to modify element vertices
      implicit none
      include 'SIZE'
      include 'TOTAL'

      real deltaT,hs,lambda
      common /myparams/ deltaT,hs,lambda

      deltaT = 2.5  !temperature change across solid domain
      hs = 0.2      !thickness of solid domain
      lambda = 10.0 !ratio of solid to fluid conductivity

      return
      end
```

Figure 5.1: The section of the conjugate heat transfer tutorial describing how to provide user data to the Nek5000 case

For the velocity and temperature profiles, these are

$$u(y) = \frac{3}{2}U_m\left(1 - 4\left(\frac{y}{H}\right)^2\right) \tag{30}$$

$$T(x,y) - T_b(x) = \frac{q''H}{2\lambda}\left(3\left(\frac{y}{H}\right)^2 - 2\left(\frac{y}{H}\right)^4 - \frac{39}{280}\right) \tag{31}$$

where the bulk temperature is given by

$$T_b(x) = \left(\frac{2q''}{U_m\rho c_p H}\right)x + T_{in} \tag{32}$$

and the required user parameters are listed in Table 5.1. Note that the given properties roughly correspond to air around room temperature. As a future update, this tutorial may be expanded to guide the user through the process of non-dimensionalizing their case.

5.2 Training

At the end of June, a virtual training session was held in response to a request from the microreactor program at Idaho National Laboratory. This was part of a larger effort to provide access to the NEAMS thermal hydraulic tools. The training was setup to provide a basic introduction to the Nek5000 code with the specific goals of teaching new users how to import third party meshes, implement basic boundary conditions, and how to visualize their results. A three hour course was put together that covered these topics with two example cases and an introduction to Gmsh. The

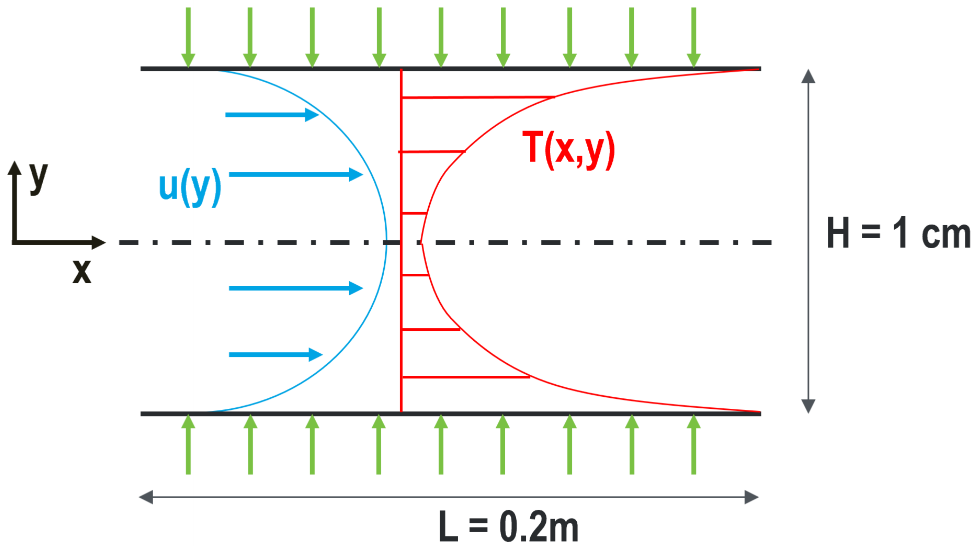


Figure 5.2: Diagram describing the case setup for fully developed laminar flow in a channel

Table 5.1: Fluid properties and simulation parameters for the fully developed laminar flow tutorial

Parameter name	variable	value	units
channel height	H	1	cm
channel length	L	20	cm
mean velocity	U_m	0.5	m/s
heat flux	q''	300	W/m^2
inlet temperature	T_{in}	10	C
density	ρ	1.2	kg/m^3
viscosity	μ	0.00002	kg/m-s
thermal conductivity	λ	0.025	W/m-K
specific heat	c_p	1000	J/kg-K

complete set of slides used for the introduction is provided in Appendix A. The training session hosted two attendees from INL and four attendees from ANL, with three "instructors" from ANL.

Prior to the start of the training, attendees were asked to make sure they can download and run Nek5000. This was facilitated by providing detailed quick-start instructions similar to those provided in NekDoc. This included details on how to setup an appropriate computing environment, e.g. which compilers to use, etc. These instructions were tuned for each lab's specific computing cluster. At INL, attendees used the Sawtooth computer. At ANL, attendees used a fully internal cluster, known as Nek5k.

The first example problem covered the most basic case setup and guided the attendees through the same laminar flow problem as described in Section 5.1. The most time was spent with this portion of the training in order to familiarize the attendees with Nek5000. Next, a primer on the use of Gmsh was given. This described how to use Gmsh for a simple pipe geometry, how to convert the resulting mesh to the Nek5000 format, and how to perform the remaining case setup through running and visualization. In addition, a few minutes were spent to show how Gmsh can be used for more complex meshes with a 4×4 rod bundle mesh and a full MSR core mesh being shown. Finally, the training session concluded with an LES case of flow inside a twisted tube, which was used to show how to use the LES model, mesh modification, and running a case dimensionlessly.

Feedback on the training was solicited from the participants both at the conclusion of the training and at two weeks after the training. Attendees indicated that the training was helpful in understanding how to use Nek5000. It was also suggested that the Gmsh portion be separated to more thoroughly cover its use and provide a more in-depth use case. This will be taken into account when planning future training sessions.

6 Summary and Future Work

This year, the Nuclear Energy Advanced Modeling Simulation program (NEAMS) thermal-hydraulics report for Nek5000 [1] NRC and verification and validation (V&V)-driven development focuses on three areas of code application and improvement. First we have continued improvements of RANS modeling capabilities in Nek5000 (and its GPU variant NekRS) including improved k-tau model focusing mostly on wall-function initial implementation with spectral element method (SEM). In addition to devising a robust way to deal with corners in SEM we have initiated an investigation of an alternative approach of eXtended/enriched spectral element method (XSEM) that greatly reduces discretization errors. We plan to continue testing and improving both approaches in the next fiscal year.

Second, we have continue assisting and collaborating with the U. S. Nuclear Regulatory Commission (NRC) staff with Nek5000 setups and validation for the Hydrogen Mitigation Experiments for REactor Safety (HYMERES-2) benchmark. In particular, after careful sensitivity study, we have settled on inlet uncertainty quantification and on simplified geometry and turbulence modeling for the full vessel geometry. The LES of long-term flow evolution in the latter geometry was obtained in low resolution while the higher resolution cases are underway together with setups of the transient heat and mass transfer cases including buoyancy effects.

We have reported on the improvement of the tet-to-hex meshing capability by implementation of a quadratic method. This method has the potential to greatly simplify the meshing procedure will providing high quality meshes that are conformal to the problem geometry. As a demonstration, it has been tested on the ROCOM pressure vessel geometry.

The documented tutorials have been revised and expanded to enhance user access to Nek5000. The conjugate heat transfer tutorial was revised for clarity and a new tutorial intended as a user's first simulation has been described. In the upcoming FY, we plan to further revise the existing tutorials and add new ones covering the use of third party meshes and the available RANS models.

Finally, a virtual training session was hosted to support the Microreactors program. This session was well-received by attendees who provided feedback on the content. This feedback will be used to improve future training sessions.

Acknowledgments

Argonne National Laboratory's work was supported by the U.S. Department of Energy, Office of Nuclear Energy, Nuclear Energy Advanced Modeling and Simulation (NEAMS) program, under contract DE-AC02-06CH11357.

We would like to thank Professor Elia Merzari and his students at Pennsylvania State University for providing the CAD file of the ROCOM facility.

We acknowledge the use of computing resources provided by the INCITE and ASCR Leadership Computing Challenge (ALCC) programs at the Argonne Leadership Computing Facility (ALCF), local computing resources provided by the NSE Division of Argonne National Laboratory, resources provided on the Bebop cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory, and the use of computing resources at Idaho National Laboratory supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517. We also acknowledge ALCF's Performance and Computation workshop focused both on simulations and data including an assistance from the workshop organizers, R. Loy et al.

References

[1] *Nek5000 Version 19.0*. December, 2019. Argonne National Laboratory, Argonne, Illinois. Available: https://nek5000.mcs.anl.gov.

[2] A. Obabko, E. Merzari, P. F. Fischer, S. M. Aithal, and A. Tomboulides, "Large-eddy simulations of stratification layer erosion by a jet," in *67th Annual Meeting of the APS Division of Fluid Dynamics*, vol. 59 (20) of *Bulletin of the American Physical Society*, (San Francisco, CA), November 2014.

[3] A. Kraus, S. Aithal, A. Obabko, E. Merzari, A. Tomboulides, and P. Fischer, "Erosion of a large-scale gaseous stratified layer by a turbulent jet – simulations with URANS and LES approaches," in *Int. Conf. on Nuclear Reactor Thermal Hydraulics (NURETH-16)*, (Chicago, IL), August 30–September 4 2015.

[4] D. Shaver, A. Obabko, A. Tomboulides, V. Coppo-Leite, Y.-H. Lan, M. Min, P. Fischer, and C. Boyd, "Nek5000 developments in support of industry and the NRC," Tech. Rep. ANL/NSE-20/48, Argonne National Laboratory, 2020.

[5] A. Tomboulides, S. M. Aithal, P. F. Fischer, E. Merzari, A. V. Obabko, and D. R. Shaver, "A novel numerical treatment of the near-wall regions in the $k-\omega$ class of RANS models," *International Journal of Heat and Fluid Flow*, vol. 72, pp. 186–199, 2018.

[6] G. Kalitzin, A. Gould, and J. Benton, "Application of two-equation turbulence models in aircraft design," *AAIA*, vol. 96, p. 0327, 1996.

[7] G. Medic, J. A. Templeton, and G. Kalitzin, "A formulation for near-wall RANS/LES coupling," *International Journal of Engineering Science*, vol. 44, pp. 1099–1112, 2006.

[8] A. Tomboulides, S. M. Aithal, P. F. Fischer, E. Merzari, and A. V. Obabko, "Initial implementation of uRANS k-omega standard, regularized and regularized SST model in Nek5000 module," Tech. Rep. ANL/MCS-TM-335, Argonne National Laboratory, July 2013.

[9] A. Tomboulides, S. M. Aithal, P. F. Fischer, E. Merzari, and A. V. Obabko, "A novel variant of the k-ω urans model for spectral element methods - implementation, verification and validation in Nek5000," in *ASME 2014 4th Joint US-European Fluids Engineering Division Summer Meeting*, vol. 1D of *ASME Proceedings, FEDSM2014-21926*, (Chicago, IL), Aug. 2014.

[10] C. G. Speziale, R. Abid, and E. C. Anderson, "Critical evaluation of two-equation models for near-wall turbulence," *AIAA Journal*, vol. 30, no. 2, pp. 324–331, 1992.

[11] D. Shaver, A. Obabko, A. Tomboulides, V. Coppo-Leite, Y.-H. Lan, M. Min, P. Fischer, and C. Boyd, "Nek5000 developments in support of industry and the NRC," Tech. Rep. ANL/NSE-20/48, Argonne National Laboratory, 2020.

[12] J. Kok and S. Spekreijse, "Efficient and accurate implementation of the $k-\omega$ turbulence model in the NLR multi-block Navier-Stokes system," Tech. Rep. NLR-TP-2000-144, National Aerospace Laboratory, 2000.

[13] "OpenFOAM version 2012." Available: www.openfoam.com.

[14] H. Grotjans and F. Menter, "Wall functions for general application cfd codes," in *ECCOMAS 98, Proceedings of the 4th European Computational Fluid Dynamics Conference, John Wiley & Sons*, pp. 1112–1112, 1998.

[15] D. Kuzmin, O. Mierka, and S. Turek, "On the implementation of the k-ε turbulence model in incompressible flow solvers based on a finite element discretisation," *International Journal of Computing Science and Mathematics*, vol. 1, no. 2-4, pp. 193–206, 2007.

[16] L. W. Ho, *A Legendre spectral element method for simulation of incompressible unsteady viscous free-surface flows.* PhD thesis, Massachusetts Institute of Technology., 1989.

[17] F. Jaegle, O. Cabrit, S. Mendez, and T. Poinsot, "Implementation methods of wall functions in cell-vertex numerical solvers," *Flow Turbulence and Combustion*, vol. 85, pp. 245–272, 2010.

[18] T. Fries and T. Belytschko, "The extended/generalized finite element method: an overview of the method and its applications," *Int J Numer Methods Eng.*, vol. 84(3), pp. 253–304, 2010.

[19] J. Larsson, S. Kawai, J. Bodart, and I. Bermejo-Moreno, "Large eddy simulation with modeled wall-stress: recent progress and future directions," *Mech Eng Rev*, vol. 3(1), 2016.

[20] B. Krank and W. A. Wall, "A new approach to wall modeling in les of incompressible flow via function enrichment," *Journal of Computational Physics*, vol. 316, p. 94–116, 2016.

[21] B. Krank, M. Kronbichler, and W. A. Wall, "A multiscale approach to hybrid rans/les wall modeling within a high-order discontinuous galerkin scheme using function enrichment," *Int J Numer Meth Fluids*, vol. 90, p. 81–113, 2019.

[22] "EVOL (project n^{o}249696) final report," tech. rep., CNRS, 2014.

[23] S. Patel, P. Fischer, M. Min, and A. Tomboulides, "An operator-integration-factor splitting (OIFS) method for incompressible flows in moving domains," Tech. Rep. ANL/ALCF-17/8, Argonne National Laboratory, 2017.

[24] G. Busco, E. Merzari, and Y. A. Hassan, "Invariant analysis of the reynolds stress tensor for a nuclear fuel assembly with space grid and split type vanes," *Int'l J. Heat and Fluid Flow*, vol. 77, pp. 144–156, Jan. 2019.

[25] H. Yuan, M. A. Yildiz, E. Merzari, Y. Yu, A. Obabko, G. Botha, G. Busco, Y. A. Hassan, and D. Thien, "Spectral element applications in complex nuclear reactor geometries: tet-to-hex meshing," *Nucl. Eng. and Des.*, vol. 357, Aug. 2019.

[26] H. Yuan, J. Solberg, E. Merzari, A. Kraus, and I. Grindeanu, "Flow-induced vibration analysis of a helical coil steam generator experiment using large eddy simulation," *Nucl. Eng. and Des.*, vol. 322, pp. 547–562, July 2017.

[27] M. A. Yildiz, Y. Hassan, H. Yuan, and E. Merzari, "Numerical simulation of isothermal flow across slant five-tube bundle with spectral element method code Nek5000," *Nucl. Tech.*, pp. 659–669, 2017.

[28] M. A. Yildiz, G. Botha, H. Yuan, E. Merzari, R. C. Kurwitz, and Y. A. Hassan, "Direct numerical simulation of the flow through a randomly packed pebble bed," *J. Fluids Eng.*, 2019.

[29] E. V. Abel, M. Anderson, and M. Corradini, "Numerical investigation of pressure drop and local heat transfer of supercritical CO2 in printed circuit heat exchangers," in *Supercritical CO2 Power Cycle Symposium*, (Boulder, CO), 2011.

[30] *High-Order Methods for Incompressible Fluid Flow.* Cambridge University Press, 2002.

[31] E. Merzari, P. Fischer, M. Min, S. Kerkemeier, A. Obabko, D. Shaver, H. Yuan, Y. Yu, J. Martinez, L. Brockmeyer, L. Fick, G. Busco, A. Yildiz, and Y. Hassan, "Toward exascale: overview of large eddy simulations and direct numerical simulations of nuclear reactor flows with the spectral element method in Nek5000," *Nucl. Tech.*, pp. 1–17, 2020.

[32] M. Kremmer, D. Bommes, I. Lim, and L. Kobbelt, "Advanced automatic hexahedral mesh generation from surface quad meshes," in *The 22nd International Meshing Rountable*, 2013.

[33] "spectral element mesh generation and improvement methods," 2016.

[34] A. Inc., "ANSYS meshing user's guide," Oct. 2012.

[35] G. Christophe and R. Jean-Francois, "Gmsh: a 3-d finite element mesh generator with build-in pre- and post-processing facilities," *Int. J. Numer. Methods and Eng.*, vol. 79, no. 11, pp. 1309–1331, 2009.

[36] L. A. Schoof and V. R. Yarberry, "EXODUS II: a finite element data model," 1994.

[37] C. Dyken and M. S. Floater, "Transfinite mean value interpolation," *Comput. Aided Geom. Des.*, vol. 26, no. 1, pp. 117–134, 2009.

[38] W. J. Gordon and L. C. Thiel, "Transfinite mappings and their application to grid generation," *Appl. Math. Comput.*, vol. 10–11, pp. 171–233, 1982.

[39] W. J. Gordon and C. A. Hall, "Construction of curvilinear co-ordinate systems and applications to mesh generation," *Int. J. Numer. Methods Eng.*, vol. 7, no. 4, pp. 461–477, 1973.

[40] D. Hermes and P.-O. Persson, "High-order solution transfer between curved triangular meshes," *arXiv Prepr.*, no. arXiv:1810.06806, pp. 1–23, 2018.

[41] S. Kliem, T. Suhnel, U. Rohde, T. Hohne, H. M. Prasser, and F. P. Weiss, "Experiments at the mixing test facility ROCOM for benchmarking CFD codes," *Nucl. Eng. and Des.*, vol. 328, no. 3, pp. 566–576, 2008.

[42] S. Kliem, T. Hohne, U. Rohde, and F. P. Weiss, "Experiments on slug mixing under natural circulation conditions at the ROCOM test facility using high-resolution measurement techniques and numerical modeling," *Nucl. Eng. and Des.*, vol. 240, no. 9, pp. 2271–2280, 2010.

[43] "Nek5000 17.0 documentation." https://nek5000.github.io/NekDoc.

[44] N. Foroozani, D. Krasnov, and J. Schumacher, "Turbulent convection for different thermal boundary conditions at the plates," *J. Fluid Mech.*, vol. 907, p. A27, Nov. 2021.

Appendix

A Training slides

U.S. DEPARTMENT OF ENERGY | Office of NUCLEAR ENERGY

Clean. Reliable. Nuclear.

Nek5000 Training
June 25th, 2021

Dillon Shaver, Haomin Yuan, Jun Fang
Argonne National Laboratory

Outline

- Introduction to Nek5000
  - Practical background on the spectral element method
- Setting up and running one of the most basic cases
  - Laminar, heated flow in a channel
- Mesh generation/importing with Gmsh
- Setting up and running a more advanced case
  - LES flow for a twisted tube

2 energy.gov/ne

Objectives

- As a result of this training, you should:
  - Be able to import meshes from third party meshing software
  - Understand how Nek5000 handles boundary conditions
  - Understand the basic case parameters used by Nek5000
  - Be able to export results to third party post-processing software (i.e. ParaView or VisIt)

3 energy.gov/ne

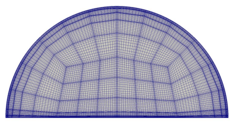
Before you begin

- This training assumes you have:
  - Good knowledge of fluid dynamics
  - Working knowledge with a Unix-based environment
  - Access to a computing cluster with at least 32 cores
    - Including installed C and Fortran compilers, and an MPI wrapper
  - Working knowledge of Fortran (e.g., do loops; if-then-else statements)

4 energy.gov/ne

Overview of Nek5000 – what you need to know

- No GUI
- Nek5000 uses the Fortran77 standard, i.e. no dynamic memory allocation
  - Each case needs to be compiled to run
  - No external dependencies
  - Compiles quickly
- Spectral elements vs finite volume
  - In the spectral element method (SEM), elements are further subdivided according to Gauss-Lobatto-Legendre (GLL) quadrature
  - Solution is defined continuously across the entire domain, rather than discretely at element centroids
  - Provides high-order spatial approximation (typically 7th order)



Cross section of a 13th order mesh, showing GLL points

Overview of Nek5000 – where to go for more information

- Documentation is available online

  https://nek5000.github.io/NekDoc/

- Very active users mailing list

  https://groups.google.com/g/nek5000

- Contact us directly!

  dshaver@anl.gov

U.S. DEPARTMENT OF ENERGY | Office of NUCLEAR ENERGY

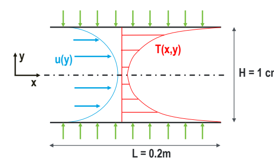
Clean. Reliable. Nuclear.

Case 1: Laminar Flow in a Channel

The introductory CFD case

Fully developed laminar flow in a channel

- Air enters a channel with fully developed velocity and temperature profiles
- A constant heat flux is applied at both walls
- Known solution makes it easy to confirm if the problem is setup correctly
- Useful engineering quantities will be determined



$$u(y) = \frac{3}{2}U_m\left(1 - 4\left(\frac{y}{H}\right)^2\right)$$

$$f = \frac{96}{Re} \qquad Nu = \frac{140}{17} \qquad T(x,y) - T_b(x) = \frac{q''H}{2\lambda}\left(3\left(\frac{y}{H}\right)^2 - 2\left(\frac{y}{H}\right)^4 - \frac{39}{280}\right)$$

Laminar flow in a channel – getting started

- Go to your scratch directory

  Sawtooth:
  ```
  $ cd /scratch/`whoami`
  ```
  Nek5k:
  ```
  $ cd /beegfs/run/`whoami`
  ```

- Unzip the necessary case files and enter the case directory

  ```
  $ tar -xzvf channel.tar.gz
  $ cd channel
  ```

- Confirm that you have `channel.usr` and `channel.par` in your case directory

Laminar flow in a channel – generate the mesh

- A simple 2D box mesh will be generated using the native Nek tool, genbox
- Create a new text file named "channel.box" with the following

channel.box
```
-2
2
Box
-50 -5
0.0 0.2 1.0
0.0 0.005 0.7
v  ,O  ,SYM,W
t  ,O  ,I  ,f
```

- Line 1: number of dimensions (negative value indicates a binary mesh file will be generated)
- Line 2: number of "fields" (i.e. velocity and temperature)
- Line 3: the geometry is a "Box"-type
- Line 4: number of elements in the x and y directions, (negative value indicates spacing will be automatically generated)
- Lines 5 and 6: minimum and maximum coordinates with the geometric growth ratio
- Line 7: velocity boundary conditions
  - In order: x_{min}, x_{max}, y_{min}, and y_{max}
  - Dirichlet, Outlet, Symmetry, Wall
- Line 8: temperature boundary conditions
  - Dirichlet, Outlet, Insulated, flux

Laminar flow in a channel – generate the mesh

- Once you have `channel.box`, the mesh can be generated using genbox

  ```
  $ genbox <<< channel.box
  ```

- This will produce the mesh file, `box.re2` which should be renamed to `channel.re2`

  ```
  $ mv box.re2 channel.re2
  ```

- Now, the map file can be generated

  ```
  $ genmap
  ```

- Genmap will ask for the mesh file and a tolerance

  `> channel` Note that the `.re2` suffix is assumed

  `> 0.2` The default tolerance is fine for this case

- You should now have

  ```
  channel.box   channel.re2   channel.ma2
  channel.usr   channel.par
  ```


Laminar flow in a channel – set the problem SIZE

- Copy the SIZE file template to your working directory

  ```
  $ cp ~/Nek5000/core/SIZE.template ./SIZE
  ```

- Open the SIZE file with a text editor and change the highlighted lines

SIZE
```
! BASIC
parameter (ldim = 2)
parameter (lx1 = 8)
parameter (lxd = 12)
parameter (lx2 = lx1 - 0)

parameter (lelg = 250)
parameter (lpmin = 1)
parameter (lelt = lelg/lpmin + 3)
parameter (ldimt = 1)
```

- These two lines correspond to the number of dimensions (`ldim`) and the number of global elements (`lelg`) for your case
- Other important parameters
  - polynomial approximation order (`lx1`)
  - Minimum number of MPI ranks (`lpmin`)
  - Number of temperature + passive scalar arrays (`ldimt`)
- Next, we'll look at the `.par` file

Laminar flow in a channel – setting the input parameters

- Open `channel.par` with a text editor

Channel.par

```
# Nek5000 parameter file
[GENERAL]
#startFrom = restart.f00000
dt = 1.0e-4
numSteps = 10000
writeInterval = 2000

userParam01 = 0.01   #channel height [m]
userParam02 = 0.5    #mean velocity [m/s]
userParam03 = 300.0  #Heat flux [W/m^2]
userParam04 = 10     #Inlet temperature [C]

[VELOCITY]
viscosity = 0.00002
density = 1.2

[TEMPERATURE]
conductivity = 0.025
rhoCp = 1200.0
```

- Properties evaluated for Air at ~20ºC
- Many of the basic parameters are readable
- The `.par` file is totally case insensitive
- The user parameters (e.g. userParam01) are a convenient method of passing extra information to Nek5000
- A more complete list with descriptions is available in the documentation
  - https://nek5000.github.io/NekDoc/problem_setup/case_files.html#parameter-file-par

Laminar flow in a channel – the user file

- The `.usr` file is used to customize the models and physics used by Nek5000
- It contains various subroutines for interfacing the solver and governing equations
  - `uservp` – variable properties
  - `userf` – momentum source term (e.g. gravity)
  - `userq` – energy/passive scalar source term
  - `userbc` – set the boundary conditions
  - `useric` – set the initial conditions
  - `userchk` – monitor the solution
  - `userqtl` – add thermal divergence (for variable density)
  - `usrdat`, `usrdat2`, `usrdat3` – general routines called during initialization
- The highlighted routines will be relevant for this case

Laminar flow in a channel – the user file

- Open `channel.usr` and scroll to `userbc`

```
 77       subroutine userbc(ix,iy,iz,iside,eg) ! set up boundary conditions
 78 c
 79 c     NOTE ::: This subroutine MAY NOT be called by every process
 80 c
 81 c      implicit none
 82
 83       integer ix,iy,iz,iside,eg
 84       real qpp
 85
 86       include 'SIZE'
 87       include 'TOTAL'
 88       include 'NEKUSE'
 89
 90 c      if (cbc(iside,gllel(eg),ifield).eq.'v01')
 91
 92       H    = uparam(1)      !channel height
 93       um   = uparam(2)      !mean velocity
 94       qpp  = uparam(3)      !heat flux
 95       Tin  = uparam(4)      !mean inlet temperature
 96       con  = cpfld(2,1)     !thermal conductivity
 97       term = qpp*H/(2*con)
 98
 99       ux   = um*3./2.*(1-4.*(y/H)**2)
100       uy   = 0.0
101       uz   = 0.0
102       temp = term*(3.*(y/H)**2-2.*(y/H)**4-39./280.)+Tin
103       flux = qpp
104
105       return
106       end
```

- Lines 95 -98 show access to the user parameters
- The highlighted lines show where the boundary conditions are set
  - `ux` – inlet velocity
  - `temp` – inlet temperature
  - `flux` – wall heat flux
- Note that care must be taken to ensure the proper boundaries are set for complex geometries

Laminar flow in a channel – the user file

- Scroll down to `useric`

```
108       subroutine useric(ix,iy,iz,eg)
109
110 c      implicit none
111
112       integer ix,iy,iz,eg
113
114       include 'SIZE'
115       include 'TOTAL'
116       include 'NEKUSE'
117
118       um   = uparam(2)
119       Tin  = uparam(4)
120
121       ux   = um
122       uy   = 0.0
123       uz   = 0.0
124       temp = Tin
125
126       return
127       end
```

- The constant mean value is set for velocity on line 121
- The constant inlet temperature is set for temperature on line 124
- More complex expressions can be used, e.g. T(x,y,z)

Laminar flow in a channel – the user file

- Scroll down to `userchk`

```
c     Evaluate friction factor
      L = 0.2
      Pin = bc_average(pr,'v  ',1)
      Pout = bc_average(pr,'O  ',1)
      darcy = -2.*Dh*(Pout-Pin)/(L*rho*um*um)
      Re = rho*um*Dh/mu
      derror = abs(1.-darcy*Re/96.)

c     Evaluate Nusselt number
      Tbulk = glsc3(t,vx,bm1,n)/glsc2(vx,bm1,n)
      Twall = bc_average(t,'f  ',2)
      HTC = qpp/(Twall-Tbulk)
      Nuss = HTC*Dh/con
      Nerror = abs(1.-Nuss*17./140.)

c     Print to logfile
      if(nio.eq.0) then
        write(*,*) "Friction factor = ",darcy,derror
        write(*,*) "Nusselt = ",Nuss,Nerror
        write(*,*)
      endif
```

- Inlet pressure, outlet pressure and wall temperature are evaluated by calling a custom function
- Bulk temperature is evaluated using built in routines for array multiplication
- The Darcy friction factor and the Nusselt number are evaluated and printed to the logfile, along with the associated error

Laminar flow in a channel – compile and run!

- Confirm you have the following in your case directory

```
SIZE          channel.par     channel.usr
channel.re2   channel.ma2
```

- Compile the case

```
$ makenek channel
```

- Submit to the queue (1 node, 0 hours, 10 minutes, 4 cores/node)

```
$ nekk channel 1 0 10 4
```


Laminar flow in a channel – While running

- Once your case starts, you will get a logfile. You can watch the case run with
  `$ tail –f logfile`
- You should see:

Current physical time

```
Step   6535, t= 6.5350000E-01, DT= 1.0000000E-04, C=  0.292 9.9066E+01 1.1849E-02
               Solving for Hmholtz scalars
         6535  Hmholtz TEMP       28   4.2002E-08   7.5188E+03   1.0000E-07
         6535  Scalars done  6.5350E-01  4.4961E-03
               L1/L2 DIV(V)          -2.0358E-10   7.6430E-08
               L1/L2 QTL              0.0000E+00   0.0000E+00
               L1/L2 DIV(V)-QTL      -2.0358E-10   7.6430E-08
         6535  Fluid done  6.5350E-01  3.7073E-03
 Friction factor =   0.15999999998656511        8.3968054731542452E-011
 Nusselt =    8.3095465134302700        9.0163623451040564E-003
```

Solver information, Ignore this for now

Laminar flow in a channel – visualization

- You should now have multiple output files:

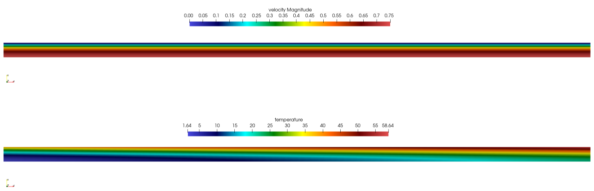
```
channel0.f00001
channel0.f00002
channel0.f00003
```

- These can be visualized in VisIt or ParaView
- Generate a metadate file with the `visnek` script

```
$ visnek
```

- Download the metadata and output files to the same folder on your local computer
- Open the metadata file with ParaView/VisIt

Laminar flow in a channel – visualization



Case 2: Using Gmsh

Importing 3rd party meshes

Gmsh to Nek5000

- Open source finite element mesh generator
- Download executable directly from http://gmsh.info/

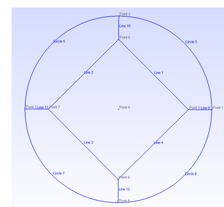
- Step by step for pipe flow
  - Mesh generation in Gmsh
  - Convert to Nek5000 mesh
  - Running in Nek5000
  - Visualizing data

Mesh generation in Gmsh

- Open Gmsh executable, open pipe.geo file
  - Sawtooth:
  - /scratch/yuanhaom/NekTraining2021/pipe_nek5000
  - Nek5k:
  - /beegfs/scratch/hyuan/NekTraining2021/pipe_nek5000

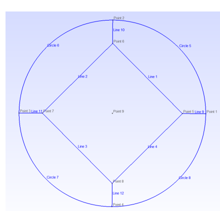
- pipe.geo
  - Meshing procedure should be scripted
  - GUI helps visualization
  - Open it both in text edit and Gmsh

- Define variables
- Define points
- Define lines
  - Based on points



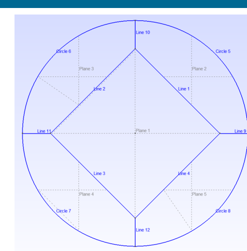
Mesh generation in Gmsh

- Open Gmsh executable, open pipe.geo file
- pipe.geo
  - Meshing procedure should be scripted
  - GUI helps visualization
- Define variables
- Define points
- Define lines
  - Based on points



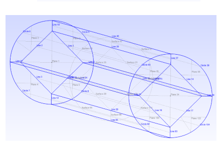
Mesh generation in Gmsh

- Define line loops
  - Based on lines
- Define surfaces
  - Based on line loops



Mesh generation in Gmsh

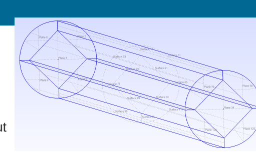
- Define 2D structured mesh
- Extrusion to 3D

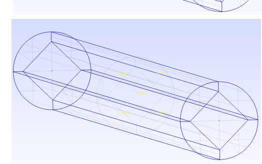


Mesh generation in Gmsh

- Define physical surfaces
  - For boundary mesh output

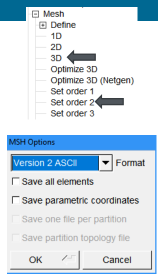
- Define physical volume
  - For volume mesh output





Mesh generation in Gmsh

- Generate 3D mesh
- Convert to 2nd order element
- Export mesh
  - File
  - Export
  - Choose "Mesh – Gmsh MSH (*.msh)"
  - Choose "Version 2"
- pipe.msh



Convert to Nek5000 mesh

- Upload pipe.msh to Blues
- Gmsh2nek: convert Gmsh .msh file to Nek5000 .re2 file
- Physical surface ID was passed to .re2 file for boundary condition set up in Nek5000.
- pipe.re2

```
[hyuan@blueslogin2 pipe]$
/lcrc/project/NEK5000Training2019/hyuan/bin/gmsh2nek
Enter mesh dimension: 3
Input .msh file name: pipe
 total node number is       13817
 total quad element number is        480
 total hex element number is       1600
 ****************************************************
 Boundary info summary
 BoundaryName    BoundaryID
 inlet          1
 outlet          2
 wall          3
 ****************************************************
 Enter number of periodic boundary surface pairs:
0

writing pipe.re2
```


Running Nek5000 simulation

- Run 'genmap'
- Set actual boundary condition in Nek5000
  - Start with zero.usr file
  - Add boundary condition to usrdat2()

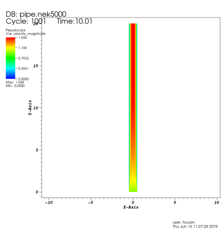
```
do iel=1,nelv
do ifc=1,2*ndim
  id_face = bc(5,ifc,iel,1)
  if (id_face.eq.1) then        ! surface 1 for inlet
     cbc(ifc,iel,1) = 'v  '
  elseif (id_face.eq.2) then    ! surface 2 for outlet
     cbc(ifc,iel,1) = 'O  '
  elseif (id_face.eq.3) then    ! surface 3 for wall
     cbc(ifc,iel,1) = 'W  '
  endif
enddo
enddo
```


Running Nek5000 simulation

- Set up SIZE file
  - lx1 : GLL points per element along each direction
  - lxd : 2*lx1/3
  - lelg : max total number of elements
  - lpmin : min MPI ranks
  - lpmax : max MPI ranks
- Compile Nek5000 executable
  - ./makenek pipe
- Running Nek5000 job
  - Running in serial
    - nek pipe
    - nekb pipe
  - Running in parallel
    - nekmpi pipe 4
    - nekbmpi pipe 4

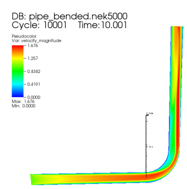
Visualizing data

- Use 'visnek' to generate a metadata file .nek5000
- open .nek5000 file in VISIT or ParaView



Advance:

- Kick on turbulence
  - Decrease viscosity
- Bended pipe



GMSH programming language

- In addition to the operations within the GUI, GMSH also supports a C-flavored programming language.
- Examples to create the geometric entities:

Point:
    Point(newp) = {x1, y1, z1, 1.0};
Edge: (has the direction)
    Line(newl) = {p1, p2}; to create a straight linel
    Circle(newl) = {p1, circle_center_id, p2}; to create a circle arc;
Face:
    Curve Loop (startSurface)={l1, l2, l3, l4};
    Surface (startSurface)={startSurface};
Volume:
    Surface Loop (startVolume) = {s1,s2,s3,s4,s5,s6};
    Volume (startVolume) = {startVolume};

GMSH programming language

- GMSH programming language offers all the standard mathematical operations:

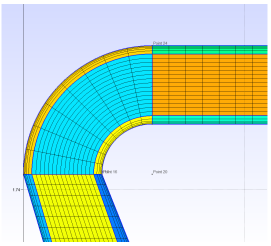
  +, -, *, /, %, Sin(_), Tan(_), Sqrt(_), etc.

```
For iedge In {pref5:pref5+3:1}
        Line(newl) = {iedge-93,iedge};
EndFor
If (pcore2 > pcore1)
        Rotate {{0,1,0}, {0,0,0}, i*Pi/8} { Point{pcore1:(pcore2-1)}; }
EndIf
Function fillSurface
        startSurface=startSurface+1;
        Curve Loop (startSurface)={l1, l2, l3, l4};
        Surface (startSurface)={startSurface};
Return
```

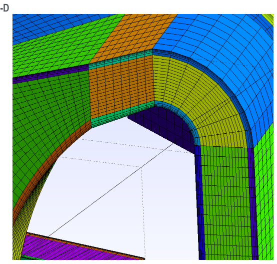
Other key functionalities:
Translation,
Rotation,
Symmetry (i.e., mirroring),
Extrusion,
Etc.,

Creating boundary layer mesh in GMSH

2-D

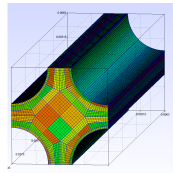


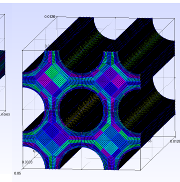
3-D

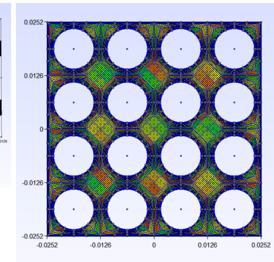


Transfinite Curve {lref2-12:lref2-1}= n_bl Using Progression ratio_bl;

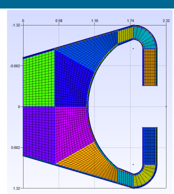
GMSH examples (fuel rod bundles)

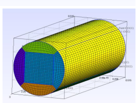


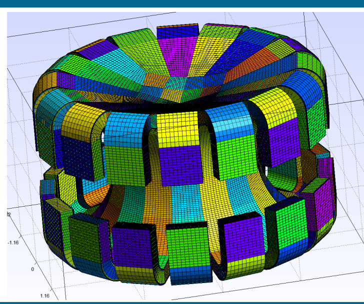




GMSH examples (continued)







GMSH caveats

- Though GMSH is a powerful meshing tool, but it was not designed to generate pure hex meshes originally.
- Creating geometric model in the programming mode involves a steep learning curve.
- GMSH is great in producing meshes for geometries of low and medium complexity, but not a suitable tool for very complex models. (thinking about how easy to divide the model into smaller blocks).
- It is very difficult to make major changes in an established GMSH model. Sometimes, it is just easier to restart from the scratch in order to make certain changes. For example, you want to try a different blocking strategy.

U.S. DEPARTMENT OF ENERGY | Office of NUCLEAR ENERGY

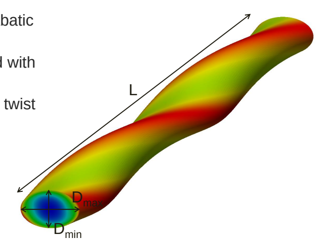
Clean. Reliable. Nuclear.

Case 3: Turbulent flow in a twisted tube

A dimensionless case

Turbulent flow in a twisted tube

- Molten salt flows through an adiabatic twisted tube
- An LES turbulence model is used with an explicit filtering method
- The tube is periodic and a full 2π twist is simulated
- The case is run dimensionlessly



Turbulent flow in a twisted tube– getting started

- Go to your scratch directory

Sawtooth:
```
$ cd /scratch/`whoami`
```
Nek5k:
```
$ cd /beegfs/run/`whoami`
```

- Unzip the necessary case files and enter the case directory

```
$ tar -xzvf twisted.tar.gz
$ cd twisted
```

- Confirm that you have

```
SIZE         twisted.par  twisted.usr
twisted.re2  twisted.ma2  restart.f00000
```


Turbulent flow in a twisted tube – setting the input parameters

- Open `twisted.par` with a text editor

Channel.par

```
#Nek5000 parameter file
[GENERAL]
startFrom = restart.f00000
dt = 2.0e-3
numSteps = 10000
writeInterval = 1000
filtering = explicit
filterWeight = 0.05
filterModes = 2

userParam01 = 1.7  #max/min diameter ratio
userParam02 = 7.5  #twist length/min diameter ratio

[PRESSURE]
residualTol = 1.0e-6
residualProj = yes

[VELOCITY]
density = 1.0
viscosity = -5000.0
residualTol = 1.0e-8
```

- The provided restart file, restart.f00000, has a turbulent velocity field
- The settings for the LES filter are highlighted, these are reasonable general settings
- Solver tolerances for pressure and velocity are specified
- Note that the value given for viscosity is the Reynolds number (highlighted)
- The negative value tells Nek5000 to treat it as a Reynolds number instead of viscosity directly

Turbulent flow in a twisted tube – the user file

- The .usr file is used to customize the models and physics used by Nek5000
- It contains various subroutines for interfacing the solver and governing equations
  - `uservp` – variable properties
  - `userf` – momentum source term (e.g. gravity)
  - `userq` – energy/passive scalar source term
  - `userbc` – set the boundary conditions
  - `useric` – set the initial conditions
  - `userchk` – monitor the solution
  - `userqtl` – add thermal divergence (for variable density)
  - `usrdat`, `usrdat2`, `usrdat3` – general routines called during initialization
- The highlighted routine will be relevant for this case

Turbulent flow in a twisted tube – the user file

- Open `twisted.usr` and scroll to `usrdat`

```
      subroutine usrdat()
      implicit none
      include 'SIZE'
      include 'TOTAL'

      param(54) = -3.0
      param(55) = 1.0

      return
      end
```

- Internal parameters are set to control the flow rate
- Used in conjunction with periodic BCs ONLY!
- Parameter 54 is set to enforce a mean velocity in the z-direction
- Parameter 55 provides the value of the mean velocity

$$U_m = 1$$

Turbulent flow in a twisted tube – the user file

- Scroll down to `usrdat2`

```
      dmni = 1.0
      dmxi = uparam(1)
      ptch = uparam(2)

      nv=nx1*ny1*nz1*nelv

      peri=pi/2.0*(3.0*(dmxi+dmni)-
     &     sqrt((3.0*dmxi+dmni)*(dmxi+3.0*dmni)))
      Ft=pi*dmxi*dmni/4.0
      dhyd=4.0*Ft/peri

      dmin=dmni/dhyd
      dmax=dmxi/dhyd
      dpch=ptch/dhyd

…

      xscale=dmax/(xmax-xmin)
      yscale=dmin/(ymax-ymin)
      zscale=2.0*pi/(zmax-zmin)

      call cmult(xm1,xscale,nv)
      call cmult(ym1,yscale,nv)
      call cadd (zm1,-zmin,nv)
      call cmult(zm1,zscale,nv)
```

- The highlighted lines are used to calculate the hydraulic diameter
- The entire domain is scaled by this factor to effectively give

$$D_h = 1$$

- The second set of highlighted lines distort the circular tube into an oval shape

Turbulent flow in a twisted tube – the user file

- Continuing in `usrdat2`

```

      do i=1,nv
        x0=xm1(i,1,1,1)
        y0=ym1(i,1,1,1)
        theta=zm1(i,1,1,1)+atan2(y0,x0)
        rr=sqrt(x0**2+y0**2)
        xm1(i,1,1,1)=rr*cos(theta)
        ym1(i,1,1,1)=rr*sin(theta)
      enddo

      zscale=dpch/(2.0*pi)
      call cmult(zm1,zscale,nv)
```

- The highlighted loop applies the twist to the tube
- The coordinates of the mesh are stored in `xm1`, `ym1`, and `zm1`
- coordinates can be modified in `usrdat2` (as long as the element Jacobians remain positive!)
- Geometry factors are recomputed between `usrdat2` and `usrdat3`

Turbulent flow in a twisted tube – compile and run!

- Confirm you have the following in your case directory

```
SIZE         twisted.par  twisted.usr
twisted.re2  twisted.ma2
```

- Compile the case

```
$ makenek twisted
```

- Submit to the queue (2 node, 1 hour, 0 minutes)

```
$ nekk twisted 2 1 0
```


Turbulent flow in a twisted tube – While running

- Once again, you can watch the case run with

```
$ tail –f logfile
```

- You should see:

```
Step   1928, t= 2.3856000E+01, DT= 1.0000000E-03, C=  0.281 3.9571E+02 1.3897E-01
             Solving for fluid
       1928  Project PRES            1.5719E-05   3.5153E-03   2.2363E+02   8   8
       1928  PRES gmres         5    9.7638E-07   6.8454E-06   1.0000E-06   3.4259E-02   4.5246E-02    F
       1928  Hmholtz VELX       6    1.7043E-09   8.4372E-01   1.0000E-08
       1928  Hmholtz VELY       6    1.7457E-09   8.0335E-01   1.0000E-08
       1928  Hmholtz VELZ       6    2.1578E-09   9.0535E-01   1.0000E-08
       1928  Volflow Z               2.3856E+01   3.7318E-02   4.1665E-05   8.7061E-01   8.7065E-01
             L1/L2 DIV(V)            2.7936E-10   1.5783E-02
             L1/L2 QTL               0.0000E+00   0.0000E+00
             L1/L2 DIV(V)-QTL        2.7936E-10   1.5783E-02
       1928  Fluid done  2.3856E+01  1.1868E-01
```

Max CFL (try to keep around 0.4)

Pressure and z-velocity solver residuals

Pressure drop (only for forced flow)

L2 Error norm in continuity (only meaningful in non-dimensional runs)

Turbulent flow in a twisted tube – visualization

- You should now have multiple output files:

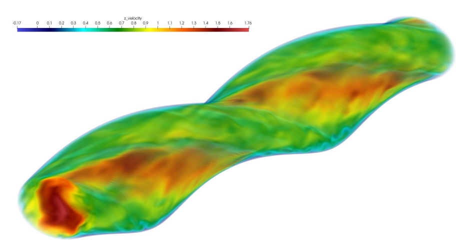
```
twisted0.f00001
twisted0.f00002
twisted0.f00003
```

- These can be visualized in VisIt or ParaView
- Generate a metadate file with the `visnek` script

```
$ visnek
```

- Download the metadata and output files to the same folder on your local computer
- Open the metadata file with ParaView/VisIt

Turbulent flow in a twisted tube – visualization




Clean. Reliable. Nuclear.
54
energy.gov/ne

Nuclear Science and Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov